PENGEMBANGAN FITUR NAGIOS UNTUK PEMANTAUAN JARINGAN BERBASIS SMS (*SHORT MESSAGE SERVICE*)

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Teknik Pada
Jurusan Teknik Informatika

oleh :

FRIMA BOBY FATRIA

10451025528

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS ISLAM NEGERI SULTAN SYARIF KASIM RIAU

PEKANBARU

2011

PENGEMBANGAN FITUR NAGIOS UNTUK PEMANTAUAN JARINGAN BERBASIS SMS (*SHORT MESSAGES SERVICE)*

FRIMA BOBY FATRIA
10451025528

Tanggal Sidang : 06 Juni 2011
Periode Wisuda : Juli 2011

Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Sultan Syarif Kasim Riau

ABSTRAK

Dalam proses keberlangsungan bisnis yang ditangani suatu jaringan harus tetap terjaga. Sangat diperlukan suatu sistem yang mampu pemantauan aktifitas jaringan selama 24 jam penuh sekaligus terintegrasi dengan alat yang mampu memberikan notifikasi saat ketersediaan akses ke jaringan sedang terputus. Dengan demikian meskipun seorang administrator berada diluar jam kerjanya, informasi mengenai gangguan pada jaringan dapat diketahui lebih awal. Nagios adalah *tools* pemantauan jaringan yang memiliki kemampuan untuk merekam statistik jaringan berdasarkan konsep SNMP dan mengirimkan pesan peringatan bila terjadi masalah di jaringan dalam bentuk *email*. Masalahnya, *email* memiliki kekurangan dalam hal akses. Oleh sebab itu, diperlukan suatu teknologi serta cara pengiriman pesan peringatan yang cepat, akurat dan handal. Integrasi Nagios dan Gammu SMS *gateway* mengirimkan *alert* notifikasi berupa SMS kepada *contact* yang telah ditentukan, dilakukan secara otomatis, dan fitur penentuan masalah jaringan yang dinamis dengan *poller* maksimal 1 menit.

Kata Kunci: *Contacts, Email*, SMS, SNMP, Nagios

DAFTAR ISI

Halaman

DAFTAR TABEL

Tabel **Halaman**

DAFTAR ISTILAH

Client = Bagian Sistem

Database = Basis data

Down = Tidak Terkoneksi

E-mail = *Electronic Mail* (Surat Elektronik)

Hardware = Perangkat keras

Hand Phone = Telepon Genggam

Host = Rekan Pengguna

Implementasi = Pelaksanaan atau penerapan

Informasi = Penerangan, pemberitahuan, kabar atau berita tentang sesuatu

Kriteria = Ukuran yang menjadi dasar penilaian atau penetapan Sesuatu

Monitoring = Pemantauan

NMS = *Network Monitoring System*

Network = Jaringan

Objektif = Mengenai keadaan yang sebenarnya tanpa dipengaruhi pendapat atau pandangan pribadi

Orientasi = Peninjauan untuk menentukan sikap yang tepat dan benar atau pandangan yang mendasari pikiran,

Open Source = Gratis biaya lisensi

Ok = Bagus

Prosedur = Tahap kegiatan untuk menyelesaikan suatu aktivitas atau metode langkah demi langkah secara pasti dalam memecahkan suatu masalah

Proses = Runtunan perubahan dalam perkembangan sesuatu

Platform = Jalur Lintasan

Port	= Tempat yang telah ditetapkan
Sistematika	= Pengetahuan mengenai klasifikasi (penggolongan)
Software	= Perangkat Lunak
SMS	= *Short Messages Service* (Layanan Pesan Singkat)
SNMP	= *Simple Network Monitoring Protocol*
Scrypts	= Kumpulan Perintah
Server	= Pusat
Service	= Layanan
Threshold	= Ambang Batas
Terstruktur	= Permasalahan yang dapat dipecahkan oleh prosedur perhitungan terkomputerisasi
Testing	= Pengujian (percobaan) untuk mengetahui tingkat kemampuan atau mengetahui mutunya
User	= Pemakai
User Interface	= Tampilan antar muka pemakai
Warning	= Peringatan

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pemantauan jaringan terdiri dari pemantauan, pengontrolan, dan perencanaan terhadap sistem atau sumber daya jaringan komputer. Pemantauan merupakan kegiatan yang dilakukan oleh seorang administrator jaringan untuk memelihara atau menjaga agar sistem dan sumber daya jaringan dapat berjalan lancar. Pemantauan jaringan komputer semakin diutamakan untuk menjaga integritas dan lebih terstrukturnya jaringan yang dibangun. Informasi mengenai komponen jaringan meliputi *hardware, software,* penanggung jawab, lokasi dan nama masing-masing *workstation/host* dapat diketahui secara jelas.

Pada sistem operasi, fasilitas yang umumnya digunakan adalah *ping*. Kelemahan dari cara pemantauan seperti ini adalah hanya memberitahu bahwa di suatu tempat antara stasiun pemantauan dan perangkat target terdapat gangguan komunikasi. Gangguan bisa jadi router, switch, atau bagian jaringan lainnya yang tidak baik, atau memang *host*-nya yang sedang *down*. Tes *ping* hanya mengatakan bahwa koneksi *down*, tidak di mana terjadinya *down*.

Sebuah penelitian pernah dilakukan untuk mengatasi beberapa masalah yang telah disebutkan di dalam paragrap sebelumnya. Pada tugas akhir (Eky Rahayu Arisanti, 2010) Fitur Cacti diangkat sebagai sebuah *tool* untuk pemantauan jaringan. Peneliti mengintegrasi fitur Cacti dengan aplikasi NowSMS *gateway* untuk notifikasi yang berbasis SMS (*Short Messages Service*). Karena fitur Cacti memiliki notifikasi yang berbasis *E-mail* pada awalnya.

Dalam proses keberlangsungan bisnis (*bussiness continuity*) yang ditangani suatu jaringan harus tetap terjaga. Sangat diperlukan suatu sistem yang mampu pemantauan aktifitas jaringan selama 24 jam penuh sekaligus terintegrasi dengan alat yang mampu memberikan notifikasi saat ketersediaan akses ke

jaringan sedang terputus. Meskipun seorang administrator berada diluar jam kerjanya, informasi mengenai gangguan pada jaringan dapat diketahui lebih awal. Bertujuan untuk mengantisipasi munculnya masalah lain yang jauh lebih besar pada jaringan.

Nagios adalah sebuah *tools* pemantauan jaringan yang bekerja untuk memantau sistem dan jaringan. Sistem berupa *resource* yang berjalan pada *server*, seperti *CPU, Memory, Disk* dan *bandwit* yang memiliki notifikasi berupa *E-mail*. Namun hingga tugas akhir ini dibuat, belum ada tambahan pada fitur Nagios berupa SMS (*Short Message Service*) untuk notifikasi administrator tentang kerusakan yang terjadi pada jaringan. Dalam tugas akhir ini akan di implementasikan suatu sistem yang dapat memonitor kinerja jaringan menggunakan Nagios sekaligus memberikan pengembangan baru berupa notifikasi berbasis SMS kepada administrator jaringan agar penanganan masalah pada jaringan dapat dilakukan sedini mungkin.

Dari beberapa keterangan diatas, fitur Nagios dapat menjadi sebuah alternatif yang lebih baik dalam melakukan pemantauan jaringan. Selain fitur Nagios mampu diintegrasikan dengan aplikasi SMS *gateway*. Proses notifikasi fitur Nagios dibagi menjadi dua kategori, yaitu *Host* dan *Service*, sehingga semua notifikasi berdasarkan *Host* dan *Service* disampaikan secara keseluruhan sesuai dengan kebutuhan admin. Nagios adalah *tools* pemantauan yang bersifat *open source* yang berlisensi GNU (*General Public Lisence)*, tidak akan memberatkan dari segi biaya lisensi. Nagios bersifat *open source,* sehingga dapat di kustomisasi sesuai kebutuhan.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang telah dibahas sebelumnya maka dapat diambil rumusan masalah yaitu bagaimana mengembangkan fitur Nagios berbasis SMS (*short messages services*) untuk notifikasi masalah jaringan.

1.3 Batasan Masalah

Batasan masalah yang akan dibuat meliputi:

1. Sistem hanya melaporkan gangguan yang terjadi kepada administrator jaringan, dalam hal ini penanganan gangguan (*troubleshooting*) masih dilakukan secara langsung ke lokasi kejadian
2. SMS server yang akan digunakan adalah Gammu SMS *Gateway*
3. Sistem hanya difokuskan untuk memonitor jaringan dengan media transmisi kabel. Jaringan dengan media transmisi nirkabel tidak termasuk dalam cakupan bahasan
4. Menggunakan kartu layanan GSM (*Global Service Mobile*)

1.4 Tujuan Penelitian

Penelitian yang dilakukan pada tugas akhir ini bertujuan sebagai berikut:

1. Mengembangkan fitur Nagios berbasis SMS (*Short Messages Service*).
2. Memonitor aktifitas jaringan serta memberikan laporan peringatan kepada administrator jaringan yang tidak sedang berada di lokasi kejadian atas terjadinya kerusakan pada jaringan
3 Mengintegrasikan dan mengkostumisasi *script Contacts*, *script Commands*, *script* Nagios, *script Host* dan *script* Gammurc sehingga Nagios berbasis SMS dapat berjalan dengan benar.

1.5 Sistematika Penulisan

Sistematika pembahasan yang akan digunakan dalam penulisan tugas akhir ini adalah:

BAB I : PENDAHULUAN

Pada bab ini menjelaskan tentang latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, sistematika penulisan.

BAB II : LANDASAN TEORI

Pada bab ini menjelaskan tentang analisa kinerja jaringan, standar keamanan jaringan dan contoh permasalahan, manajemen jaringan, *management information bases*, Arsitektur SNMP (*Simple Network Monitoring System*), Protokol SNMP, Perkembangan SNMP.

BAB III : METODOLOGI PENELITIAN

Pada bab ini menjelaskan tentang pengumpulan data, analisa, perancangan, implementasi, pengujian, kesimpulan dan saran.

BAB IV : ANALISA DAN PERANCANGAN

Pada bab ini menjelaskan tentang analisa info kebutuhan admin, analisa sistem baru.

BAB V : IMPLEMENTASI DAN PENGUJIAN

Pada bagian ini menjelaskan tentang implementasi sistem dan pengujian sistem.

BAB VI : PENUTUP

Pada bagian ini menjelaskan tentang kesimpulan dan saran.

BAB II

LANDASAN TEORI

Jaringan adalah kumpulan dari sejumlah perangkat berupa komputer, hub, switch, router, atau perangkat jaringan lainnya yang terhubung dengan menggunakan media komunikasi tertentu (Wagito, 2005). Perangkat yang terhubung dengan jaringan disebut juga sebagai *node*. Hal ini memungkinkan pengguna dapat bertukar dokumen dan data, mencetak pada printer yang sama, dan menggunakan sumber daya jaringan (*hardware* dan *software)* ada.

Sebuah jaringan komputer biasanya terdiri dari 2 buah komputer atau lebih dan melakukan data *sharing* antar komputer. Informasi dan data bergerak melalui media komunikasi. Media komunikasi yang dipakai dalam membuat jaringan komputer antara lain adalah kabel, jaringan telepon, gelombang radio, satelit, bluetooth atau infra merah. Pemakaian media komunikasi ini akan tergantung pada kegunaan dan ukuran jaringan.

2.1 Analisa Kinerja Jaringan

Analisa kinerja jaringan komputer membicarakan sifat dasar dan karakteristik aliran data, yaitu efisiensi daya-kerja, penundaan dan parameter lainnya yang diukur untuk dapat mengetahui bagaimana suatu pesan diproses di jaringan dan dikirim lengkap sesuai fungsinya.

Analisa kinerja jaringan komputer dapat didefinisikan sebagai penelitian kuantitatif yang terus menerus terhadap suatu jaringan komunikasi dalam urutan kerja yang tetap berada dalam fungsinya (Terplan, 1992) agar:

1. Dapat menyempurnakan level layanan pemeliharaan.
2. Dapat mengenali potensi kemacetan

3. Dapat mendukung pengendalian operasional jaringan, administrasi dan merencanakan kapasitas

Administrasi jaringan membantu langkah analisa kinerja dalam usaha mengevaluasi kemampuan layanan pada konfigurasi tertentu, selanjutnya akan mendefinisikan indikator kinerja yang penting, merekomendasikan prosedur pelaporan kinerja.

2.1.1 Kategori

Kategori Analisa:

1. Analisis kinerja dengan tujuan Optimalisasi Sistem dalam layanan yang cepat, tepat dan akurat.
2. Analisis kinerja dengan tujuan Optimalisasi Sistem dalam bidang keamanan sistem, data dan informasi, yang sering dikenal dengan istilah *Penetration Test* yaitu dengan cara melakukan penyelidikan terhadap sistem dari sudut pandang si penyerang. Tujuan utamanya adalah untuk mengidentifikasi temuan dan resikonya sebelum mencari suatu solusi.
3. Analisis Hybrid, analisis keseluruhan terhadap berbagai potensi sistem yang dapat ditingkatkan kinerjanya dengan tujuan evaluasi dan pengembangan sistem.

2.1.2 Parameter Kinerja Jaringan

Kriteria penting dari sudut pandang pemakai jaringan adalah keandalan, yaitu kriteria pengukuran seberapa mudah suatu sistem terkena gangguan, terjadi kegagalan atau beroperasi secara tidak benar. Kehandalan adalah ukuran statistik kualitas komponen dengan menggunakan strategi pemeliharaan, kuantitas redudansi, perluasan jaringan secara geometris dan kecenderungan statis dalam merasakan

sesuatu secara tidak langsung tentang bagaimana suatu paket ditansmisikan oleh sistem tersebut. Kinerja jaringan dapat diukur berdasarkan kriteria Terplan (1992):

1. Kriteria level pemakai (*user level*), yaitu waktu respon dan keandalan
    a. Waktu respon yaitu waktu tanggapan saat paket dipancarkan dengan benar.
    b. Kehandalan yaitu suatu keadaan yang dapat menentukan seberapa berfungsinya sistem pada suatu tugas pengiriman paket
2. Kriteria level jaringan (*network Level*), yaitu waktu respon rata-rata.
   Penentuan waktu respon rata-rata dilakukan dengan 2 langkah, yaitu:
    a. Menentukan rata-rata penundaan satu jalur paket melewati jaringan dan antar mukanya sebagai suatu fungsi beban terhadap ukuran paket.
    b. Menggunakan informasi dengan penundaan dan pemakaian link untuk menghitung waktu respon rata-rata pemakai
3. Kriteria kinerja khusus, yaitu daya kerja dan penundaan rata-rata.

2.1.3 SNMP Sebagai Protocol Kinerja Jaringan

Jaringan komunikasi (termasuk jaringan komputer) tidak bisa dikelola bila indikator kinerjanya tidak dapat dipantau dan diukur dengan tepat. Syarat utama rancangan jaringan adalah menetapkan level layanan pemeliharaan untuk memuaskan pengguna sebagai basis analisis kinerja. Indikator kinerja haruslah dapat menunjukkan keadaan kinerja jaringan yang dipantau. Contoh suatu indikator kinerja adalah aplikasi SNMP (*Simple Network Management Protocol*), yang didalamnya terdapat MIB (*Management Information base*) yaitu struktur database variabel elemen jaringan yang dikelola yang dikelompokkan berdasarkan parameter layanan dan parameter efisiensi.

2.1.4 Mesin Pemantauan Jaringan

Mesin pemantauan jaringan merupakan sebuah perangkat lunak atau program yang digunakan untuk mencari dan memberikan informasi tentang jaringan.

Jaringan komputer dan sistem yang terdapat didalamnya selalu terdapat kelemahan dan kekurangan. Hal yang lazim dilakukan adalah menemukan kekurangan tersebut untuk kemudian memperbaikinya. Oleh sebab itu, Nagios diberi tambahan fitur baru berbasis sms agar kekurangan yang ada dapat tertutupi.

2.2 Standar Keamanan Jaringan

Standar keamanan jaringan dimaksudkan mengenali komponen-komponen yang harus diberikan perlindungan agar tercapai suatu lingkungan jaringan yang aman dengan kebutuhan minimum sebagai berikut(Sysneta, 2010):

1. Semua titik-2 yang bisa diakses secara fisik dan juga perangkat jaringan (*routers, server* dan LAN *switches*) haruslah aman secara fisik. Jangan sampai *infrastructure* vital (missal *server room*) bisa diakses oleh sembarang orang, bahkan bila perlu disamarkan tanpa ada yang tahu fungsinya.
2. *Operating system* dan *firmware* piranti haruslah diperkuat (di*pathced*/di *update*) untuk mencegah titik-titik lemah keamanan (*security hole*)
3. Piranti-2 jaringan seperti *switches* dan *router* haruslah mempunyai *password* yang sangat kuat, *password* yang tidak umum dan tidak gampang ditebak.
4. Akses *remote* kepada piranti jaringan (Telnet) haruslah di control dengan membatasi aksesnya menggunakan sistem filter IP *address* kepada *remote device* yang memang diberikan akses saja dan hanya oleh *personal* IT *support* saja.
5. Piranti jaringan haruslah mempunyai "*Message of the Day* (MOTD)" atau *banner login* yang mendefinisikan *Warning* pesan Legal setiap kali diakses,

dengan pesan larangan kepada semua pengguna yang tidak terotorisasi jika mencoba untuk mengakses piranti jaringan tersebut.

6. *Session time-out* pada *console* dan telnet haruslah di setting dan dibatasi tidak boleh lebih dari 10 menit saat *idle* kepada semua piranti jaringan. Hal ini untuk menjaga pelanggaran keamanan jika terminal tersebut ditinggal dalam keadaan masih logon.
7. *Password* dan nama *community* SNMP haruslah paling sedikit 8 karakter dan haruslah terdiri dari *alphanumeric password*. *Password* harus tidak gampang ditebak.
8. Manajemen *services* seperti SNMP haruslah di non aktifkan jika tidak dipakai
9. Semua komunikasi *public* (internet dan *wireless*) haruslah di enkripsi. Enkripsi haruslah secara regular diganti dengan cara yang aman untuk menjaga pengupingan dan serangan manipulasi data.
10. Pertahanan parameter haruslah ditekankan pada segenap titik jaringan yang menghadap ke *public* termasuk internet. Hal ini bisa dilakukan dengan menggunakan paket *filtering* (*extended access-list*). Untuk koneksi ke internet sebuah *firewall* dengan konfigurasi *policy* yang sangat kuat haruslah diterapkan. Gunakan *policy* yang sangat ketat untuk *inbound traffic* dari internet dengan *extended access-list* pada semua parameter *router*. *Access-list* haruslah simple dan sangat efektif dalam mengontrol *traffic* yang tidak diinginkan dan memberikan keamanan kuat kepada asset penting.

Selain itu, seorang pakar teknologi jaringan David Icove mengklasifikasikan tingkat keamanan yang harus diperhatikan meliputi (Yodi, 2010):

1. Fisik/*Physical Security*
2. Manusia/*Personel Security*

3. Data, media, teknik dan komunikasi

4. Kebijakan dan prosedur

Penjelasan masing-masing tingkat keamanan tersebut terhadap keamanan jaringan adalah untuk tingkat keamanan secara fisik adalah memastikan semua komponen fisik jaringan terpasang secara benar. Keamanan fisik lebih dipandang pada sisi *hardware* dari jaringan komputer serta peralatan pendukungnya, seperti AP (*Access Point*), Kabel LAN (*Local Area Network*), *cable tray, Chasing CPU, UPS, AC* untuk ruangan *server* dan lain sebagainya.

Keamanan terhadap personel (*personel security*) yang menggunakan jaringan komputer perlu diperhatikan. Sebagai contoh kasus Kevin Mitnick (Hacker Legendaris), dimana dia mampu menjebol sistem keamanan perusahaan dengan cara berpura-pura menjadi pegawai *service* komputer. Hal yang mungkin terjadi adalah sistem keamanan jaringan komputer rusak gara-gara orang asing yang masuk ataupun personel yang bekerja dalam jaringan itu sendiri. Bila terkait dengan orang asing, keamanan bisa dibentuk dengan membuat sistem *ID-CARD*, penempatan petugas keamanan ataupun penerapan level personel. Bagaimana bila orang dalam? Sangat sulit membentuk standar keamanan personel karena ini berkaitan dengan etika dan moral dari personel itu sendiri. Satu-satunya harapan adalah HRD mampu menyeleksi pegawai yang jujur dan bisa dipercaya dalam bekerja.

Keamanan pada bidang data, media, teknik dan komunikasi cenderung mendapatkan perhatian yang lebih daripada yang lain. Keamanan pada sisi ini menitikberatkan pada aspek *software*. Data pada jaringan tentu harus diamankan dari berbagai pihak yang tidak memiliki izin khusus. Dalam jaringan komputer dapat ditentukan medianya untuk berkomunikasi menggunakan media dan teknik tertentu. Contoh kasusnya, komunikasi transfer file dilakukan dengan sistem *sharing*. Ada pemberian ijin kepada komputer tertentu yang bisa mengakses data-data penting. Dalam jaringan tersebut, menggunakan media *sharing* file sistem seperti NFS (*Network File System*) dalam pertukaran data. Secara teknik, setiap komputer harus

mengetahui alamat komputer lain bila hendak melakukan *sharing*. Dalam gambaran umum tersebut, keamanan ditingkatkan dengan cara menggunakan *password* dalam pengaturan hak pengaksesan data. Selain itu, komputer lain tidak boleh menggunakan media lain selain sistem *file sharing* dalam berkomunikasi.

Kebijakan (*policy*) serta prosedur dalam jaringan komputer harus ditentukan secara tegas guna membangun sistem keamanan yang kuat. Penggolongan beberapa pengguna jaringan komputer terhadap kepentingannya haruslah diperhatikan. Aplikasinya, bila pada suatu jaringan komputer perusahaan, terdapat 3 bidang kerja yaitu keuangan, *customer service* dan produksi. Pengaturan kebijakan dan prosedur diterapkan berdasarkan kepentingan bidang kerja masing-masing. Contohnya, perusahaan tersebut mendapatkan koneksi internet. Perlu dilihat bahwa kepentingan divisi keuangan yang tidak membutuhkan internet dalam operasionalnya. Sedangkan divisi *customer service* dan produksi membutuhkan akses internet dalam operasionalnya. Maka pengaturan kebijakan disini adalah koneksi internet cuma diberikan pada jaringan komputer divisi *customer service* dan produksi sedangkan divisi keuangan tidak diberikan akses internet. Namun secara total, jaringan komputer divisi keuangan, divisi *customer service* serta divisi produksi saling terhubung satu dengan yang lain.

Penjelasan panjang mengenai keamanan jaringan diatas bukan sekedar wacana, akan tetapi terlebih dahulu memang wajib diterapkan pada jaringan komputer untuk meminimalisir resiko yang akan terjadi. Bersamaan dengan hal itu, penggunaan sistem pemantauan juga sangat perlu diterapkan guna memberikan jaminan ketersediaan akses ke jaringan karena tujuan utama sebuah sistem pemantauan adalah memastikan bahwa jaringan komputer selalu *available* (tersedia). Jika hal tersebut tidak dimungkinkan, maka tujuan utama selanjutnya adalah memastikan bahwa informasi ketidaksediaan jaringan tersebut dapat diperoleh dengan cepat.

Ketidak sediaan dalam jaringan sering dikaitkan dengan berbagai masalah yang terjadi di jaringan. Masalah yang terdapat pada jaringan dapat dikelompokkan dalam dua kategori; yakni masalah jaringan yang berhubungan dengan konektifitas dan masalah yang berkaitan dengan kinerja jaringan (Ekklesya, 2009).

Masalah konektifitas terjadi ketika suatu *End Station* (seperti komputer, hub/*switch*, router) tidak dapat berkomunikasi satu sama lain baik di lingkungan jaringan LAN (*Local Area Network*) maupun WAN (*Wide Area Network*). Masalah konektivitas meliputi *Lose of Connectivity* dan *Intermittent Connectivity*

a. Kehilangan Konektifitas (*Lose Of Connectivity*)

Lose of Connectivity (*LoC*) adalah suatu kondisi yang terjadi dimana para pengguna layanan jaringan tidak dapat mengakses jaringan sama sekali. Penyebab terjadinya masalah ini bisa bermacam-macam, akan tetapi secara umum masalah ini hampir dapat dipastikan karena terjadinya kesalahan dan kerusakan pada peralatan fisik di jaringan. Sebagai contoh; kabel yang rusak, peralatan pada *server* farm misalkan router yang tiba-tiba mati karena terlalu panas, kesalahan konfigurasi routing pada router sehigga menyebabkan koneksi gagal dengan status *ip acquiring,* dan sebagainya.

b. Konektivitas Terputus – Putus Atau Tidak Stabil (*Intermittent Connectivity*)

Kondisi ini terjadi apabila para pengguna memiliki akses kesumber daya jaringan beberapa kali tetapi mereka kadang kala masih menghadapi koneksi jaringan yang "mati" (*Periods Of Downtime*). Masalah konektivitas yang terputus dapat mengindikasikan bahwa jaringan berada pada ambang kerusakan yang lebih parah. Masalah ini pada beberapa kasus berkemungkinan dapat disebabkan oleh permasalahan fisik yang tidak baik (misalkan voltase yang tidak stabil sehingga transfer data terputus-putus), sementara pada beberapa kasus lainnya disebabkan oleh masalah kinerja yang akan dibahas selanjutnya.

Masalah kinerja jaringan merupakan masalah yang timbul pada jaringan ketika ia tidak dapat beroperasi secara efektif sesuai dengan sumber daya dan beban kerja yang ada. Sebagai contoh, waktu respon jaringan tidak stabil seperti biasanya, dan para pengguna banyak yang mengeluh bahwa jaringan melayani pekerjaan mereka lebih lama. Beberapa masalah kinerja yang muncul seperti duplikasi alamat dan laju utilisasi jaringan yang selalu konsisten tinggi dapat saja terjadi. Contoh lain yang dapat mengganggu kinerja jaringan adalah terjadinya down pada suatu titik di jaringan karena lalu lintas paket pada suatu waktu melebihi kapasitas *bandwidth*, kapasitas *log server* yang terlalu penuh, kapasitas *hardware* dan memori yang sudah tidak lagi memadai karena meningkatnya aktifitas jaringan, transfer data yang tiba-tiba melambat yang bisa saja dikarenakan pertambahan jumlah *user* yang tidak terotorisasi.

Masalah kinerja jaringan yang buruk dan tidak segera diperbaiki akan menimbulkan masalah lain yang berdampak jauh lebih besar terutama pada keberlangsungan bisnis yang ditangani oleh jaringan. Oleh sebab itu untuk menangani masalah inipun diperlukan cara dan prosedur yang tepat (Ekklesya, 2009).

2.3 Manajemen Jaringan

Dengan berkembangnya jaringan TCP/IP yang sangat pesat, maka diperlukan juga suatu manajemen untuk mengatur jaringan. *Internet Architecture Board* (IAB) merekomendasikan RFC 1052 yang berisikan tentang Simple *Network Management Protocol* (SNMP), ISO *Common Management Information Service*/*Common Management Information Protocol* (CMIS/CMIP), IAB menyarankan untuk menggunakan SNMP.

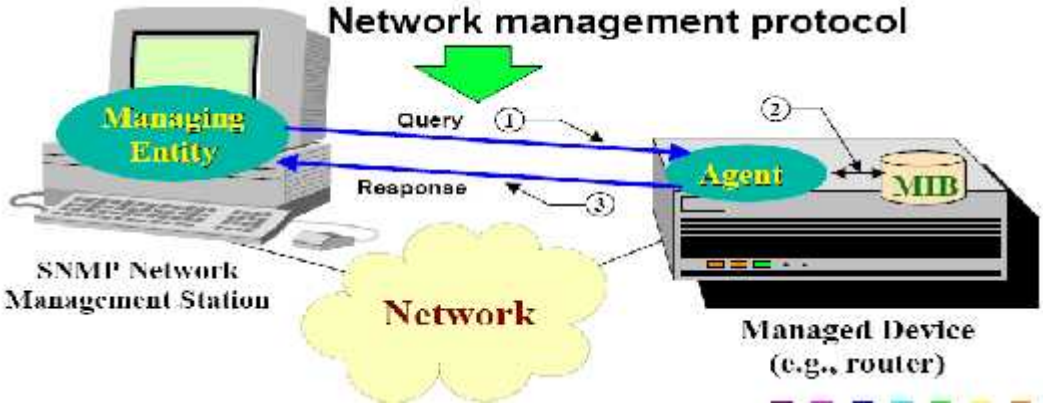
2.3.1 *Simple Network Management Protocol* (SNMP)

SNMP (protokol manajemen jaringan yang bersifat simpel) merupakan salah protokol resmi dari Internet Protocol suite yang dibuat oleh *Internet Engineering Task Force* (IETF). SNMP merupakan contoh dari layer 7 aplikasi yang digunakan oleh *network management* sistem untuk memonitor perangkat jaringan sehingga dapat memberikan informasi yang dibutuhkan bagi pengelolanya.

2.3.2 Konsep SNMP

SNMP digunakan untuk me-manage perangkat yang berada di dalam internet menggunakan protokol TCP/IP. SNMP menyediakan sekumpulan operasi dasar untuk memantau (*monitoring*) dan me-*maintain* internet. SNMP menggunakan konsep *manager* dan *agent*. *Manager* (biasanya berupa suatu *host*) mengendalikan dan memantau sekumpulan *agent*.

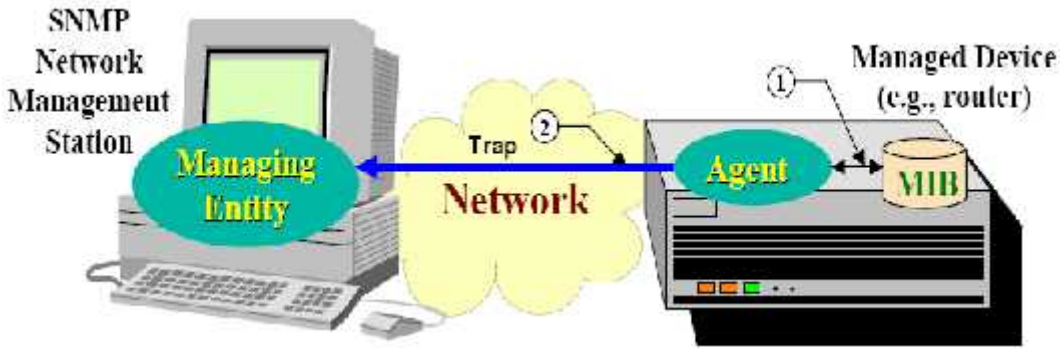
Management station (disebut manajer) merupakan suatu *host* yang menjalankan program *SNMP client. Managed station* (disebut *agent*) adalah router atau *host* yang menjalankan program *SNMP server.* Manajemen jaringan diperoleh melalui interaksi antara *manager* dengan *agent*.



Gambar 2.1 SNMP *Manager* meminta Informasi pada SNMP *Agent*

Gambar 2.1 menjelaskan interaksi yang terjadi antara *management station* (manajer) dengan *agent*. Untuk mendapatkan informasi yang terdapat pada database *agent* (MIBs), *manager* akan mengirimkan permintaan berupa *query* pada *agent*. *Agent* memiliki dan menyimpan informasi *performance* di dalam suatu *database* yang disebut MIB (*Management Informatin Based*). *Agent* akan berinteraksi dengan MIB dan memproses *query*.

Setelah *query* diproses, *agent* akan mengiriman respon berupa *trap*. *Trap* adalah pesan yang dikirimkan dari *agent* ke *manager* sebagai laporan dari suatu event. Gambar 2.2 dibawah ini mendeskripsikan tentang proses pengiriman *trap*.



Gambar 2.2 SNMP *Agent* Memberikan Response Berupa *Trap* Pada SNMP *Manager*

Manager dapat mempunyai akses terhadap nilai (*value*) yang terdapat di dalam *database* tersebut. Misalnya sebuah router menyimpan informasi variabel jumlah paket yang diterima dan jumlah paket yang di-*forward*; *manager* dapat mengambil dua informasi tersebut lalu membandingkannya untuk mengambil keputusan apakah router tersebut sedang berada di dalam kondisi kongesti atau tidak

Manager dapat pula menyuruh router (*host*) untuk melakukan aksi tertentu. *Agent* dapat pula berkontribusi terhadap proses manajemen. Program *server* yang berjalan di *agent* dapat mengecek kondisi router dan bila ditemukan gejala yang tidak biasa, maka *agent* dapat mengirimkan pesan peringatan (*warning message*) yang disebut *trap* ke *manager*

Dengan kata lain, proses manajemen SNMP menggunakan tiga ide dasar:

1. Suatu *manager* mengecek suatu *agent* dengan cara meminta informasi kelakuan *agent*
2. Suatu *manager* dapat memaksa *agent* untuk melaksanakan tugas tertentu dengan me-*reset* nilai (*value*) di dalam *database agent*
3. *Agent* berkontribusi terhadap proses manajemen dengan memperingatkan *manager* bila terdapat situasi yang tidak biasa.

2.3.3 *Management Information Base* (MIBs)

MIB merupakan *database* yang digunakan untuk manajemen perangkat pada jaringan. *Database* tersebut berisikan objek *entiti* dari perangkat jaringan (seperti router atau switch). Objek pada MIB didefinisikan menggunakan *Abstract Syntax Notation One* (ASN 1), dan diberi nama "*Structure of Management Information Version* 2 (SMIv2). *Software* yang digunakan untuk *parsing* disebut MIB *compiler*.

RFC yang membahas antara lain RFC1155 – Structure *and identification of Management Information for* TCP/IP *base internets*, RFC1213 – *Management Information Base for Network Management of TCP/IP-based internets*, dan RFC 1157 – A *Simple Network Management Protocol*.

SNMP, komunikasi yang terjadi antara *management station* (contoh: *console*) dengan *management object* (seperti *router*, *gateway* dan *switch*), menggunakan MIB. *Component* yang berkerja untuk mengambil data disebut SNMP *agent*, merupakan *software* yang dapat berkomunikasi dengan SNMP *Manager*.

2.3.4 Arsitektur SNMP

Framework dari SNMP terdiri dari:

1. *Master Agent*

Master agent merupakan perangkat lunak yang berjalan pada perangkat yang mendukung SNMP, dimana bertujuan untuk merespon permintaan dari SNMP dari *management station*. *Master agent* kemudian meneruskan kepada *subagent* untuk memberikan informasi tentang manajemen dengan fungsi tertentu.

2. *Subagent*

Subagent merupakan perangkat lunak yang berjalan pada perangkat yang mendukung SNMP dan mengimplementasikan MIB. *Subagent* memiliki kemampuan:

a. Mengumpulkan informasi dari objek yang di*manage*
b. Mengkonfigurasi informasi dari objek yang di*manage*
c. Merespon terhadap permintaan manajer
d. Membangkitkan alarm atau *trap*

3. *Management Station*

Management station merupakan *client* dan melakukan permintaan dan mendapakan *trap* dari SNMP *server*.

2.3.5 Protokol SNMP

PDU dari SNMP (versi 1) antara lain:

1. *GET REQUEST* – digunakan untuk mendapatkan informasi manajemen
2. *GETNEXT REQUEST* – digunakan secara iteratif untuk mendapatkan sekuen dari informasi manajemen
3. *GET RESPONSE*

4. *SET* – digunakan untuk melakukan perubahan terhadap subsistem
5. *TRAP* – digunakan untuk melakukan pelaporan terhadap subsistem manajemen

Untuk versi berikutnya ditambahkan PDU:

1. *GETBULK REQUEST* – iterasi yang lebih cepat untuk mendapatkan informasi
2. *INFORM – acknowledge* terhadap *TRAP*.

2.3.6 Perkembangan SNMP

2.3.6.1 SNMP Version 1

RFC untuk SNMP, dikenal dengan nama *Simple Network Management Protocol* version 1, pada tahun 1988:

1. RFC 1065 – *Structure and identification of management information for TCP/IP-based internets*
2. RFC 1066 – *Management information base for network management of TCP/IP-based internets*
3. RFC 1067 – *A Simple Network Management Protocol*

   Kemudian menjadi kadaluwarsa dengan digantikan dengan:
4. RFC 1155 – *Structure and identification of management information for TCP/IP-based internets*
5. RFC 1156 – *Management information base for network management of TCP/IP-based internets*
6. RFC 1167 – *A Simple Network Management Protocol*

Versi 1 memiliki kelemahan pada sistem autentifikasi karena mengirimkan *password* secara *plain text*.

2.3.6.2 SNMP Version 2

Versi 2 ini banyak yang tidak menggunakan dikarenakan ketidak cocokan *framework*. *Simple Network Management Protocol version* 2 (RFC 1441 – RFC 1452) dan juga dikenal sebagai SNMP v2. Diperkenalkan *GETBULK* sebagai alternatif dari *GETNEXT*. Dikenalkan juga *Community-Based Simple Network Management Protocol version* 2 atau yang disebut SNMP v2c sebagai pengganti sistem authentifikasi *User-Based Simple Network Management Protocol version* 2, atau SNMP v2u yang digunakan untuk memperbaiki keamanan dari SNMP v1.

2.3.6.3 SNMP Version 3

Versi ini didefinisikan pada RFC 3411 – RFC 3418 yaitu *Simple Network Management Protocol version* 3, dikeluarkan pada tahun 2004. Pada prakteknya SNMP bisa menggunakan versi SNMPv1, SNMPv2c, atau SNMPv3. Dijabarkan pada RFC 3584 – *Coexistence between Version* 1, *Version* 2, *and Version* 3 *of the Internet-Standard Network Management Framework*.

2.3.6.4 Contoh Penggunaan SNMP

Beberapa contoh penggunaan SNMP:

1. Pemantauan waktu penggunaan suatu perangkat (*sysUpTimeInstance*)
2. *Inventory* dari versi sistem operasi (*sysDescr*)
3. Mengkoleksi informasi suatu *interface* (*ifName, ifDescr, ifSpeed, ifType,ifPhysAddr*). Mengukur *throughput interface* dari jaringan (*ifInOctets, ifOutOctets, ifInErrors, ifOutErrors*), yakni jumlah *total byte* yang diterima dan jumlah *total byte* yang dikirim, *ifInErrors* mendefinisikan jumlah paket

diterima yang dibuang karena rusak, *ifOutErrors* mendefinisikan jumlah paket dikirim yang dibuang karena rusak

4. Menarik informasi *cache* dari ARP (*ipNetToMedia*)

2.4 Nagios

Nagios merupakan sebuah sistem dan aplikasi pemantauan jaringan yang diciptakan oleh Ethan Galstad (http://nagios.org) dan pertama diluncurkan tahun 1999. Nagios mengawasi *host-host* dan servis yang telah ditetapkan, memberi peringatan jika keadaan memburuk, dan memberi tahu kapan keadaan tersebut membaik. Nagios dijalankan dalam Linux.

Nagios sebuah program yang memonitor *host* dan layanan di jaringan Anda, memberitahu Anda ketika masalah sedang berlangsung. Nagios bisa menjalankan sebuah *script*, dan memberikan sebuah *interface web* untuk menampilkan status sistem terkini.

Nagios dapat dikembangkan dan dapat memonitor status dari semua peristiwa jaringan. Ia melakukan cek dengan menjalankan sebuah *script* kecil dengan interval reguler, dan membandingkan hasilnya dengan hasil yang seharusnya di peroleh. Ini dapat memberikan cek yang lebih canggih dari pada sebuah sistem jaringan sederhana. Misalnya, ping mungkin akan memberitahu anda bahwa mesin sedang berjalan, dan NMAP mungkin melaporkan bahwa sebuah *port* TCP merespon pada sebuah permintaan, tetapi Nagios dapat mengambil halaman web atau membuat sebuah *query*/permintaan *database*, dan memverifikasi bahwa respon tersebut bukan sebuah kesalahan.

Nagios bisa memberitahukan sebuah peringatan awal tentang suatu masalah jaringan, seringkali memperbolehkan anda untuk merespon kepada masalah sebelum *user* punya kesempatan untuk mengadu.

Keistimewaan Nagios:

1. Pemantauan servis jaringan (SMTP, POP3, HTTP, NNTP, PING, dsb)
2. Pemantauan sumber *host* (*load* prosesor, penggunaan disk, dsb)
3. Desain *plugin* yang sederhana, yang mengijinkan pengguna untuk lebih mudah menggunakan pemeriksaan terhadap *services*.
4. *Service* cek yang paralel
5. Pemberitahuan ketika terjadi masalah pada servis atau *host* dan
6. Kemampuan untuk mendefinisikan kejadian yang ditangani selama servis/ *host* berlangsung untuk mempermudah pemecahan masalah
7. Perputaran *file log* yang otomatis
8. Mendukung implementasi pemantauan dengan *host* yang berlebih
9. *Web interface* yang dinamis untuk melihat status *network*, urutan masalah dan pemberitahuan, *log* dan *file*.

Kebutuhan awal yang harus dipenuhi oleh sistem sebelum instalasi Nagios adalah sebagai berikut:

1. *Web Server* (apache)
2. PHP (php 5, php-mysql php-snmp)
3. Net-SNMP

Beberapa aktifitas pemantauan yang dapat dilaporkan oleh Nagios:

1. Nagios dapat digunakan untuk menginventori semua peralatan/*device* yang *connect* ke *network*, meski begitu Nagios yang sudah ada harus dikustomisasi (*patch* dan programnya harus diatur sedemikian rupa sesuai kebutuhan)

2. Pemantauan *Link* dan *traffic* serta aktifitas pada jaringan
3. Untuk mendapatkan info *network interface*.
4. Kapasitas partisi *hardisk*
5. Beban *processor*
6. Penggunaan memori
7. *Space of disk in server*, ruang penyimpanan dalam *server*
8. *CPU utilization* (kinerja CPU)

2.4.1 Data *Source* Nagios

Untuk menghandel pengumpulan data, dapat dibuat eksternal *script* atau *command* yang akan diperlukan untuk di pilih, Nagios kemudian menyimpan nya kedalam *database*. Data *source* Nagios mempunyai *format* CFG. Data *Sources* dapat juga di buat, yang berkoresponden dengan data sebenarnya dalam *map*. Sebagai contoh jika ingin membuat *map* ketika *ping* ke suatu *host*, harus dibuat data *sources* dengan memanfaatkan *script* yang meng-*ping* suatu *host* yang menghasil kan nilai dalam *milliseconds*. Setelah itu dapat didefinisikan informasi tambahan yang dibutuhkan oleh data *input*. Seperti *host* yang akan di *ping* dalam hal ini. Setelah data *source* di buat, lalu akan di maintain sesuai dengan waktu yang kita tetapkan secara otomatis.

2.4.2 Klasifikasi *Alert* Notifikasi Pada Nagios

Pengumpulan data pada Nagios untuk ditampilkan dalam bentuk *map* dapat dilakukan secara *real time*. Setiap data yang dikumpulkan akan dapat disimpan dalan *database* untuk kemudian dilakukan pengukuran kembali sesuai waktu yang diinginkan, jika ada masalah akan mengeluarkan notfikasi berupa *alert*, pada Nagios klasifikasi *alert* ada 3:

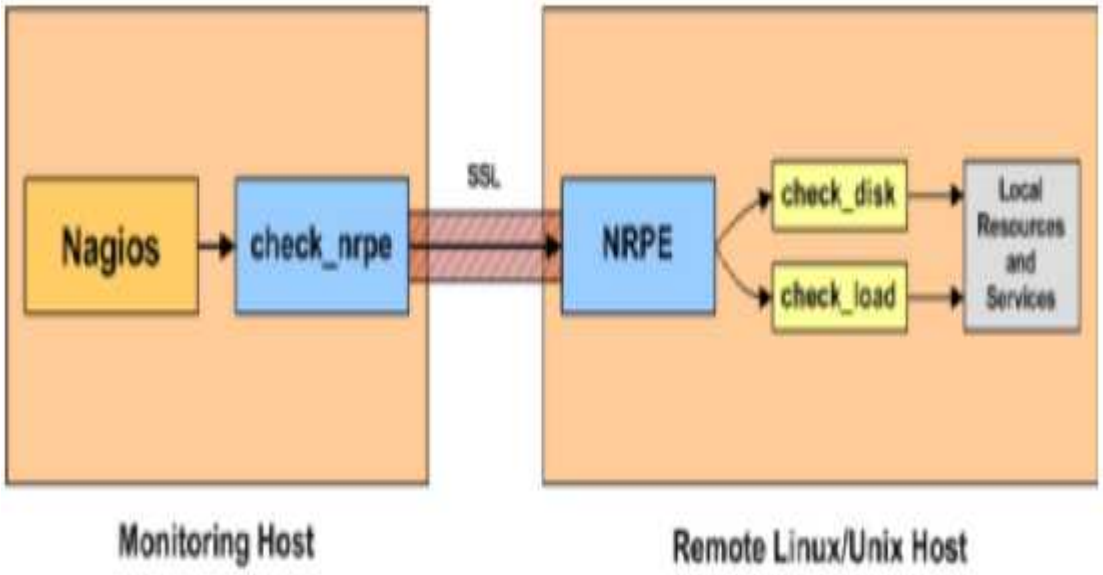
1. *Down* adalah *Host* atau *Service* dalam keadaan *down*, dan keterangan *'check time out'*.
2. *Ok* adalah *Host* atau *Service* dalam keadaan baik dan terkoneksi dengan baik.
3. *Warning* adalah *Host* atau *Service* dalam keadaan melebihi ambang batas yang kita tetapkan.

2.4.3 *User Management* (Manajemen Pengguna)

Nagios memiliki fungsi yang sangat banyak terhadap jaringan, maka manajemen berbasis *user* sengaja ditambahkan untuk menjaga otoritas terhadap *user* yang memiliki hak akses. Hal ini akan mengakibatkan seseorang yang memiliki otoritas mampu merubah parameter.

2.4.4 Nsclient++ NRPE

Nsclient++ NRPE adalah aplikasi *client* yang harus di-*install* kedalam sistem operasi *Host*. Yang dibutuhkan untuk memonitor *local resource/attribute* pada *host*, seperti penggunaan *hard disk, CPU, memory* dan lainnya. Untuk proses instalasinya akan dijabarkan pada lampiran C.



Gambar 2.3 Contoh Sistem Nsclient++ NRPE

2.5 SMS (*Short Message Service*)

2.5.1 Pengertian SMS

SMS *(Short Message Service)* adalah kemampuan untuk mengirim dan menerima pesan yang terbatas besarnya (pesan singkat) antar *handphone*/telepon selular yang berupa data dalam bentuk *string* atau teks dan data *binary*. Teks dapat terdiri dari kombinasi kata-kata, nomor-nomor, atau penggabungan huruf dan angka. Setiap SMS dibatasi hanya sampai 160 karakter saja, dengan menggunakan huruf latin, sedangkan untuk karakter non latin seperti karakter Arab atau *Chinese*, SMS dibatasi hanya sampai 70 karakter saja. *Output* pada MS (*Mobile Station*) dari data yang diterima dalam bentuk teks adalah teks juga, sedangkan *output* dari data *binary* bisa berupa teks, gambar maupun suara (*mailbox*).

SMS termasuk salah satu dari GSM data *services* yang disediakan oleh PLMN (*Public Land Mobile Network*). Contoh GSM data *services* yang lain adalah *fax* dan data *transfer*. GSM data *services* hanya dapat digunakan jika sebuah PLMN

menyediakan fasilitas tersebut.SMS pertama dikirim pada bulan Desember 1992, dari sebuah PC (*Personal Computer*) ke sebuah *handphone* pada Vodafone GSM *network* di Inggris.

2.5.2 Fasilitas Dasar SMS

Ada dua macam fasilitas dasar pada SMS, yaitu:

1. SM MT (*Short Message Mobile Terminated*)
2. SM MO (*Short Message Mobile Originated*)

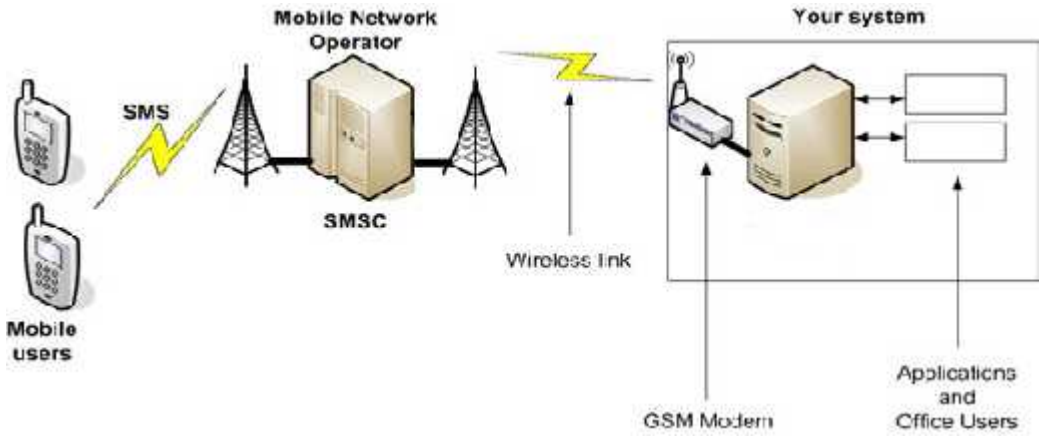
SM MT merupakan kemampuan untuk mentransfer sebuah pesan singkat yang dikirim dari SC ke salah satu MS, dan menyediakan informasi tentang pengiriman pesan tersebut apakah berhasil atau gagal.

SM MO adalah kemampuan dari sistem GSM untuk mentransfer pesan yang dikirim dari MS (*Mobile Station*) ke SME melalui SC, dan juga menyediakan informasi mengenai pengiriman pesan, apakah gagal atau sukses, juga pesan tersebut mengikutsertakan alamat dari SME supaya SC dapat berhasil mengirimkan pesan tersebut.

Sebuah MS yang aktif dapat menerima pesan singkat (SMS *deliver*) setiap saat, sebuah laporan selalu dikirimkan ke SC untuk memberi informasi bahwa MS (*Mobile Station*) sudah menerima pesan, berhasil diterima atau tidak, serta alasannya.

2.5.3 PDU (*Protocol Data Unit*)

PDU atau *Protocol Data Unit* adalah bahasa yang digunakan SMS. Data yang mengalir ke/dari SMS-*centre* harus berbentuk PDU. PDU berisi bilangan-bilangan heksadesimal yang mencerminkan bahasa I/O. PDU terdiri dari beberapa *header*. *Header* untuk mengirim SMS ke SMS-*centre* berbeda dengan SMS yang diterima dari SMS-*centre*. Maksud dari bilangan heksadesimal adalah bilangan yang terdiri atas 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F.



Gambar 2.4 Jaringan Sederhana Teknologi SMS

2.6 Gammu SMS *Gateway*

Gammu adalah aplikasi yang digunakan sebagai alat untuk memungkinkan pengembangan interaktif cepat aplikasi dan layanan SMS. Ketika Gammu menerima pesan SMS, dapat dikonfigurasi untuk mengirim pesan ke sebuah *script* yang menjalankan pada *server* HTTP. Gammu ini menyediakan cara sederhana untuk mendapatkan pesan yang diterima ke dalam sebuah aplikasi, sehingga aplikasi dapat melakukan pemrosesan kustom pada pesan, aplikasi ini juga dapat menghasilkan jawaban yang sederhana kembali ke pesan yang diterima.

2.7 Siklus Hidup Pengembangan Sistem (*System Development Life Cycle*)

Pengembangan sistem informasi berbasis komputer dapat merupakan tugas kompleks yang membutuhkan banyak sumber daya dan dapat memakan waktu berbulan-bulan bahkan bertahun-tahun untuk menyelesaikannya. Proses pengembangan sistem melewati beberapa tahapan dari mulai sistem itu direncanakan sampai dengan sistem tersebut diterapkan, dioperasikan dan dipelihara.

Di *System Development Life Cycle* (SDLC) tiap-tiap bagian dari pengembangan sistem dibagi menjadi beberapa tahapan kerja. Tiap tahapan memiliki karateristik tersendiri. Tahapan utama siklus hidup pengembangan sistem dapat

terdiri dari tahapan perencanaan (*system planning*), analisis sistem (*system analysis*), desain sistem (*system design*), seleksi sistem (*systems selection*), implementasi sistem (*system implementation*) dan perawatan sistem (*system maintenance*). (Jogiyanto, 1999)

Menurut James Taylor (Taylor,2004) tahapan SDLC meliputi tahapan sebagai berikut:

Tabel 2.1 *System Development Life Cycle*

No	*Phase*	*Activities*
1.	Konsep	Mendefenisikan Kebutuhan produk, membangun analisa kemungkinan yang akan terjadi, mendefenisikan *scope* (batasan) produk, membangun sistem arsitektur
2.	Kebutuhan (*requirement*)	
3.	Desain	menyelesaikan kebutuhan produk, melengkapi desain secara utuh (*preliminary design*)
4.	Implementasi	Meminta persetujuan desain dan tanda tangan dari pihak investor, membangun desain secara detail dan membangun sistem
5.	Integrasi dan tes	Membangun dan menghubungkan unit-unit dalam sistem, dan uji coba pengintegrasian sistem. Pengiriman sistem (*Deliver system*).
6.	Instalasi sistem	Instalasi dan uji coba sistem
7.	Pemeliharaan *(Maintenance)* dan pendukung *(support)*	Pengoperasian dan pemeliharaan sistem

Fase *concept* dan *requirement* secara garis besar merupakan tahap awal pengembangan sistem yang meliputi pendefenisian kebutuhan produk, mendefenisikan *scope* (batasan) produk serta membangun rancangan arsitektur sistem.

Tahap *design* merupakan suatu proses dimana analisa kebutuhan suatu produk telah benar-benar diketahui dan pada tahap ini desain sistem secara utuh harus dipersiapkan.

Setelah sistem di desain, tahap berikutnya adalah implementasi sistem. Dalam tahap ini, hasil desain sistem akan dibangun secara detail. Hasil implementasi akan diintegrasikan dengan keseluruhan bagian sistem. Dan tahap *testing*/pengujian akan membuktikan keabsahan sistem. Tahap terakhir dari siklus hidup sistem adalah tahap pemeliharaan. Tujuan pemeliharaan adalah agar kesinambungan sistem terjaga.

Sebelum sistem baru dibuat, perencanaan perlu dibuat untuk mendefenisikan kebutuhan. Tahap perencanaan meliputi investigasi awal untuk menentukan *project scope, project objective* dan *project methodology*. *Project scope* (batasan project) yaitu cakupan projek yang akan dibuat. *Project objective* (objek dari suatu project) muncul karena adanya permasalahan terhadap sistem yang telah ada. Metodologi *project* (*project methodologi*) yang merupakan pedoman tentang bagaimana dan apa yang harus dikerjakan selama pengembangan sistem.

Analisa kebutuhan sistem mencakup pengumpulan dan analisa informasi yang dibutuhkan untuk pengembangan sistem selanjutnya. Hal yang dilakukan pada tahap ini yaitu:

1. Identifikasi kembali masalah (*redefine the problem*)
2. Mengidentifikasi dan memahami sistem yang telah ada
3. Identifikasi dan mendefenisikan kebutuhan pengguna dan hambatan yang mungkin ditemui pada sistem baru

Tahap perancangan merupakan tahapan dimana kebutuhan suatu sistem telah jelas teridentifikasi, sehingga pada bagian ini akan dihasilkan *prototipe* dari sistem yang akan dibuat. Tahapan yang dilalui meliputi:

1. Identifikasi secara jelas kebutuhan sistem
2. Identifikasi *hardware* dan *software* yang dibutuhkan dalam implementasi sistem
3. Perancangan *interface*
4. Perancangan *database*

Setelah itu, tahap implementasi akan mengkonversikan *prototipe* dan hasil perancangan sebelumnya dalam bahasa pemrograman. Dalam tahap inilah, sistem informasi benar-benar dibangun. Tahap implementasi juga mencakup instalasi dan *coding.*

Pengujian merupakan tahap terpenting setelah sistem informasi selesai dibuat. Tahap pengujian dilakukan dengan tujuan untuk menjamin sistem yang dibuat sesuai dengan hasil analisis dan perancangan serta menghasilkan satu kesimpulan apakah sistem tersebut sesuai dengan yang diharapkan

Hasil pengujian yang telah berhasil akan dianalisa kembali pada bagian analisa akhir untuk mendapatkan hasil yang benar-benar *valid* dan akurat. Kegagalan pada tahap pengujian akan berakibat lambatnya penyelesaian suatu sistem karena sistem akan dianalisa ulang dari awal untuk mengetahui kesalahan identifikasi.

2.8 System Development Methodology

System Development Methodology (Metodologi pengembangan sisem) adalah suatu kerangka kerja yang digunakan untuk menyusun, merencanakan dan mengontrol proses pengembangan suatu sistem informasi. Metodologi adalah kesatuan metode-metode, prosedur-prosedur, konsep-konsep pekerjaan, aturan-aturan dan postulat-postulat yang digunakan oleh suatu ilmu pengetahuan, seni atau disiplin lainnya. (Jogiyanto,1999)

Menurut Jogiyanto (1999) metodologi pengembangan sistem ada 3 sebagaimana dijelaskan pada sub bab berikut:

2.8.1 *Functional Decomposition Methodologies*.

Metodologi menekankan pada pemecahan dari sistem ke dalam sub sistem yang lebih kecil, sehingga akan lebih mudah untuk dipahami, dirancang dan diterapkan. Yang termasuk adalah kelompok ini adalah:

a. HIPO (*Hierarchy Plus Input Process Output*)

b. *Stepwise Refinement* (SR) atau *Iterative Stepwise Refinement* (ISR)

2.8.2 Data *Oriented Methodologies*

Metodologi ini menekankan pada karakteristik dari data yang akan diproses. Metodologi ini dikelompokkan lagi kedalam dua kelas, yaitu:

2.8.2.1 Data *Flow Oriented Methodologies*

Metodologi ini secara umum didasarkan pada pemecahan dari sistem ke dalam modul-modul berdasarkan tipe elemen data dan tingkah laku logika modul tersebut didalam sistem. Dalam metodologi ini, sistem secara logika dapat digambarkan secara logika dari arus data dan hubungn antar fungsinya di dalam modul di sistem. Yang termasuk dalam metodologi ini adalah:

a. SADT (*Structured Analysis and Design Techniques*)

b. *Composite design*

c. *Structured Systems Analysis and Design* (SSAD)

2.8.2.2 Data *Structure Oriented Methodologies*

Metodologi ini menekankan pada struktur *input* dan *output* di sistem. Struktur ini kemudian akan digunakan sebagai dasar struktur dari sistemnya. Hubungan fungsi antar modul atau elemen sistem kemudian dijelaskan dari struktur sistemnya. Yang termasuk metodologi ini adalah:

a. JSD (*Jackson's system development*).

b. W/O (*warnier/Orr*).

2.8.2.3 *Prescriptive Methodologies*

Yang termasuk metodologi ini adalah:

a. ISOS (*Informatin System and Optimation System*).

b. PLEXSYS

   Kegunaan PLEXSYS adalah untuk melakukan transformasi suatu *statement* bahasa komputer tingkat tinggi ke suatu *executable code* untuk suatu konfigurasi perangkat keras yang diinginkan.

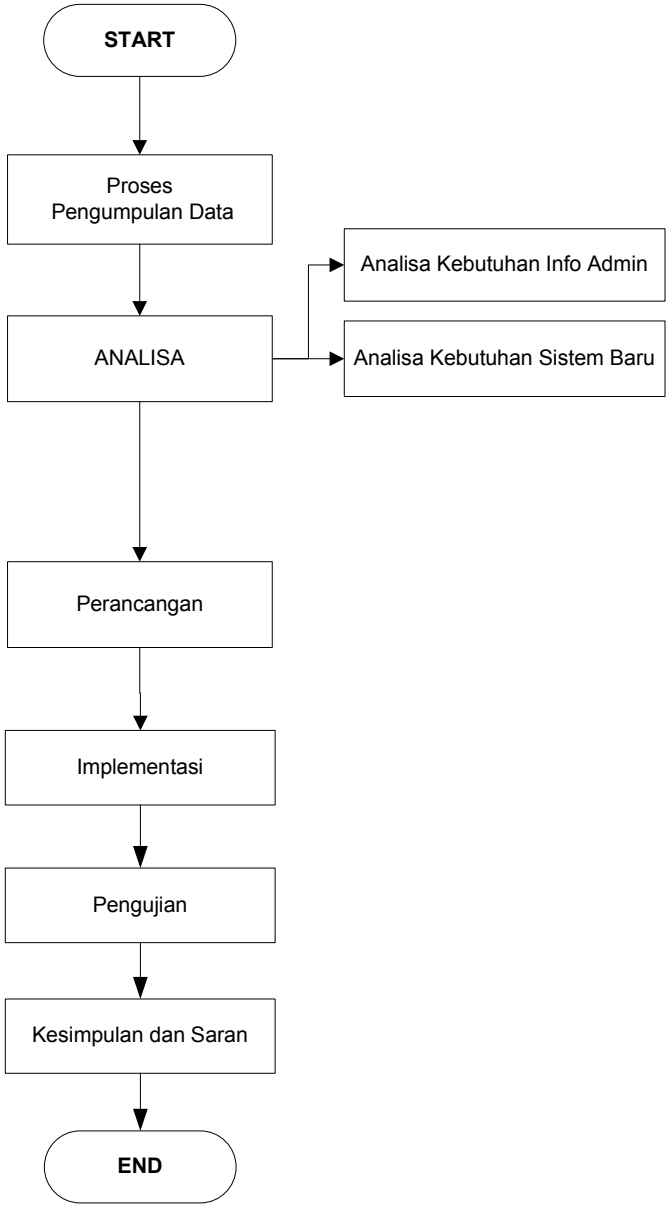
c. PRIDE

   Ditawarkan oleh suatu perusahaan di Amerika Serikat yaitu M Bryce & *Associates*.

BAB III

METODOLOGI PENELITIAN

Metodologi penelitian merupakan sistematika tahapan penelitian yang dilakukan selama pembuatan tugas akhir. Tahapan yang dilakukan tersebut tertuang pada gambar 3.1 berikut ini.



Gambar 3.1 Diagram Pengembangan Sistem

Metodologi penelitian yang akan dilakukan berdasarkan Gambar 3.1 meliputi beberapa prosedur-prosedur pengerjaan dan secara garis besar melalui enam tahapan yaitu:

3.1 Pengumpulan Data

Pengumpulan data merupakan tahapan persiapan yang harus dilaksanakan terlebih dahulu sebelum dilakukan penelitian. Berikut merupakan aktivitas yang dilaksanakan dalam pengumpulan data:

1. Studi Pustaka

   Studi pustaka berfungsi untuk mendukung penelitian yang akan dilaksanakan. Pengumpulan teori-teori yang mendukung dalam penelitian ini merupakan kegiatan dalam studi pustaka. Teori-teori bersumber dari buku, jurnal dan penelitian-penelitian sejenis.

2. Wawancara

   Wawancara berfungsi untuk mengumpulkan informasi yang akan berguna untuk tahap analisa dan tahap-tahap selanjutnya. Wawancara dilakukan terhadap praktisi jaringan

3.2 Analisa

Tahapan selanjutnya adalah melakukan analisa. Analisa merupakan suatu proses yang berguna untuk melakukan pemilahan pada apa yang akan dikerjakan. Kemudian dilanjutkan pada perancangan sistem berdasarkan dari analisa permasalahan sebelumnya. Proses ini terbagi menjadi tiga tahapan:

3.2.1 Analisa Kebutuhan Info Admin

Bagian awal dari Bab Analisa membahas tentang kebutuhan info admin yakni Nagios yang masih murni tanpa tambahan fitur. Bagian ini mencakup semua hal yang menjadi bagian dari instalasi, penggunaan, cara kerja Nagios, proses *alert* serta kelemahan dan kekurangan *alert* yang dikeluarkan Nagios.

Kemudian, disini juga mencakup bagaimana solusi yang mungkin untuk mengatasi kekurangan tersebut.

3.2.2 Analisa Kebutuhan Sistem Baru

Setelah sistem lama dianalisa, maka akan muncul kebutuhan untuk membangun sistem baru, kebutuhan itu mengacu pada apa yang menjadi kekurangan pada keluaran *alert* yang dihasilkan Nagios. Analisa kebutuhan yang dimaksud meliputi analisa kebutuhan data, analisa kebutuhan perangkat, analisa dan kebutuhan fungsi.

3.3 Perancangan

Tahap perancangan sistem merupakan prosedur untuk mengkonversi spesifikasi logis kedalam sebuah perancangan yang dapat diimplementasikan pada perangkat lunak. Perancangan yang akan dibuat meliputi perancangan basis data dan perancangan antarmuka. Keduanya dikembangkan berdasarkan hasil analisa yang telah dilakukan

3.4 Implementasi

Implementasi pengembangan sistem ini dilaksanakan dengan menggunakan bahasa pemrograman PHP dan *database* My SQL. Pada implementasi berisi juga tentang alasan pemilihan perangkat lunak beserta batasan implementasi dan lingkungan implementasi. Kebutuhan untuk membangun perangkat lunak adalah:

1. Kebutuhan Perangkat Keras (*Hardware*)

   Perangkat keras yang dibutuhkan untuk membangun sistem yaitu:

   a. *Personal computer* dengan prosesor Pentium 4 atau diatasnya

   b. 512 MB RAM

   c. *Hard disk* dengan kapasitas 40 GB

   d. Switch

e. Modem, berupa perangkat telepon genggam GSM beserta *simcard*

f. Kabel Data USB Untuk Hp Nokia N73

2. Kebutuhan Perangkat Lunak *(Software Requirements)*

Pemilihan *software* untuk tugas akhir ini akan melibatkan banyak aspek yaitu:

a. *Software* tersebut harus memiliki *user interface* yang menarik

b. *Software* tersebut harus bisa berinteraksi dengan *database*

c. *Software* tersebut harus berbasis web dan bersifat *open source*

Beberapa *hardware* dan *software* dibutuhkan untuk menunjang pengembangan sistem serta eksekusi sistem yang efisien, sistematis dan efektif. Table 3.1 menunjukkan *software* yang dibutuhkan untuk membangun sistem. Berikut ini adalah daftar *software* yang dibutuhkan untuk menunjang implementasi sistem:

Table 3.1 : Kebutuhan Perangkat Lunak untuk Membangun Sistem

Software	***Purpose***
Nagios dan *software* pendukungnya, yaitu: Net SNMP,	*Software* pemantauan yang akan diinstalasi
Microsoft Office Visio 2003	Digunakan untuk menggambar diagram.
GAMMU sms *gateway*	*Software* SMS *Gateway* yang akan diinstalasi
PHP 5	Ini adalah bahasa pemrograman yang akan digunakan untuk membangun sistem
Apache Server	Sebagai *web server* untuk menjalankan sistem.
Linux Ubuntu 10.10	Sistem operasi tempat sistem diimplementasikan
Microsoft Words 2003	Sebagai *platform* yang digunakan untuk mendokumentasikan seluruh pekerjaan
Mozilla Fire Fox	Sebagai *web browser* untuk menjalankan sistem.

Setelah dilaksanakan implementasi, maka akan dilakukan pengujian terhadap implementasi yang telah dilaksanakan dan tinjauan ulang terhadap unjuk kerja sistem.

3.5 Pengujian

Pada tahapan pengujian ini menggambarkan kondisi-kondisi yang terjadi apabila aplikasi dijalankan. Standar pengujian yang dilakukan berkaitan dengan uji fitur dan hasil pengiriman SMS notifikasi, apakah sesuai dengan yang diharapkan. Tahap pengujian adalah sebagai berikut:

1. Pengujian Proses *Login* pada Nagios
2. Pengujian Pendaftaran *Host* pada Nagios
3. Pengujian Pendaftaran *Contact*
4. Pengujian Gammu SMS *gateway*
5. Pengujian Pengiriman notifikasi SMS pada fitur Nagios
6. Pengujian Waktu *Poller* Nagios

3.6 Kesimpulan dan Saran

Hasil dari pengujian akan dikaji dan dianalisa ulang untuk mendapatkan hasil yang benar-benar diharapkan. Kesimpulan, kekurangan dan saran pengembangan sistem akan diletakkan pada bagian kesimpulan dan saran.

.

BAB IV

ANALISA DAN PERANCANGAN

Analisa adalah suatu proses identifikasi permasalahan dari kumpulan data yang bernilai informasi dan bermanfaat pada pembangunan sistem. Analisa merupakan langkah awal dalam membuat suatu aplikasi. Analisa dilakukan untuk memahami persoalan sebelum melakukan tahap perancangan. Hal ini dilakukan untuk mencari kebutuhan-kebutuhan yang diperlukan oleh sistem dan kendala-kendala yang akan dicari solusinya. Setelah analisa dilakukan, tahap selanjutnya adalah perancangan sistem. Perancangan yang dibuat harus memiliki kesesuaian dengan analisa sistem yang sebelumnya telah dilakukan.

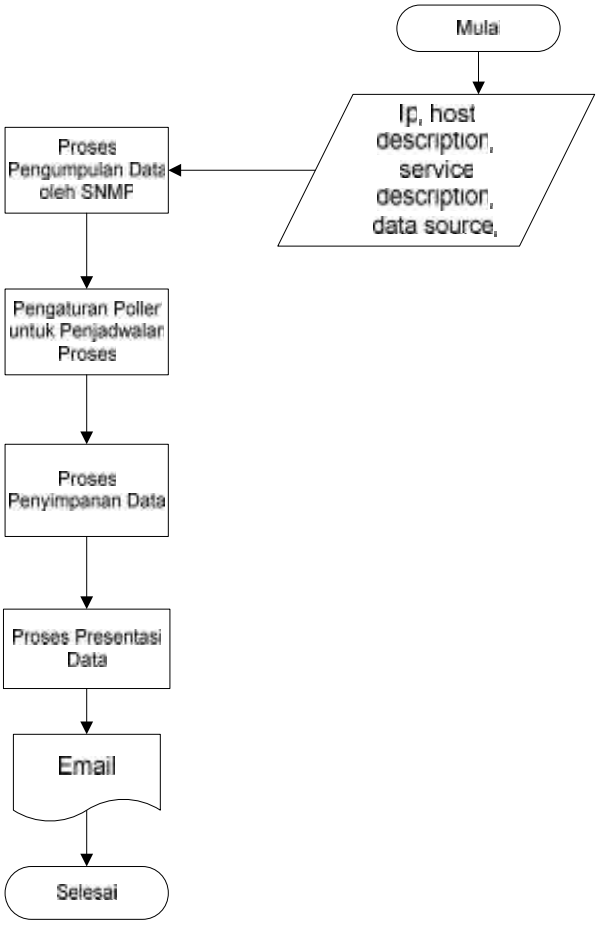
4.1. Analisa Kebutuhan Info Admin

Nagios merupakan salah satu *Network Monitoring System* yang bekerja berdasarkan konsep SNMP (*Simple Network Management Protocol)* pada pengumpulan informasi statistik jaringan. Proses pengambilan data (lewat SNMP atau skrip) sampai kepada pembuatan grafik dilakukan menggunakan bahasa pemrograman PHP.

Kebutuhan awal sebelum instalasi Nagios yaitu *Web Server* (*apache*), PHP (php 5), *Net-SNMP*. Untuk proses instalasi Nagios dapat dilihat selanjutnya pada Lampiran A.

Secara umum yang dilakukan oleh Nagios adalah mengumpulkan data. Data dikumpulkan dengan *Poller* yang dieksekusi oleh sistem operasi. *Interval* pengumpulan data atau dengan kata lain eksekusi *Poller* dapat diatur melalui fasilitas penjadwalan yang tersedia di sistem operasi. Data yang telah tersedia di *host* atau *remote* target didapatkan dengan SNMP. Sehingga tiap perangkat yang dapat menjalankan fungsi SNMP (*managed agents/nodes*) dapat dipantau secara bersamaan oleh Nagios.

Dalam perkembangannya, Nagios mampu memberikan keluaran dalam bentuk *email*. *Email* yang dihasilkan merupakan sebuah pesan peringatan akan adanya suatu kerusakan ataupun masalah dalam jaringan. Secara umum proses pemantauan oleh Nagios dapat dilihat pada gambar 4.1.



Gambar 4.1 *Flowchart* Proses Pengumpulan Statistik Jaringan Oleh Nagios

Layaknya sistem pemantauan lainnya, Nagios memiliki banyak kelebihan sekaligus kekurangan. Kelebihan Nagios antara lain:

1. Dari segi biaya, Nagios bersifat *open source* (gratis biaya lisensi) sehingga modul Nagios dapat dengan mudah diunduh dan digunakan oleh siapa saja.
2. Dibandingkan dengan MRTG yang juga bersifat *open source*, tampilan Nagios jauh lebih menarik dan fiturnya lebih lengkap.

3. Sebagai sistem pemantauan, Nagios cukup handal untuk dapat memantau aktifitas dan peralatan yang ada di jaringan

Walaupun Nagios memiliki banyak fitur dengan berbagai kelebihan, tetap saja Nagios masih memiliki keterbatasan, misalkan untuk kasus yang membutuhkan reliabilitas (kehandalan) tinggi yaitu perlu adanya ketersediaan informasi saat adminstrator berada dimanapun. Nagios belum mampu untuk menangani trouble shooting jaringan secara langsung. Nagios adalah *tools* untuk pemantauan dan belum menyertakan fasilitas untuk *trouble shooting* masalah jaringan dalam modulnya. Oleh sebab itu, tugas akhir ini akan mencoba menutupi kekurangan Nagios yakni penambahan fitur Nagios dalam hal penyediaan informasi yang lebih *fleksibel* dan *reliabel*, pembahasan mengenai hal tersebut akan dijabarkan dalam analisa selanjutnya.

4.2. Analisa Sistem Baru

Akan dibangun fitur baru untuk Nagios yang akan menyempurnakan fitur yang telah ada. Bagian ini akan menganalisa permasalahan pada Nagios sebelum ditambahkan fitur baru berupa SMS.

4.2.1. Analisa Permasalahan Nagios Tanpa Fitur SMS

Banyak hal yang menjadi alasan mengapa Nagios perlu penyempurnaan fitur. Pada bagian ini akan dianalisa permasalahan yang ada pada Nagios yang lama untuk kemudian dijadikan sebagai acuan pada analisa sistem yang akan dikembangkan.

Seorang administrator jaringan memiliki tugas yang amat berat terkait pengumpulan data statistik jaringan apabila dilakukan secara *manual*. Terlebih untuk kategori jaringan komputer yang memiliki peran penting yang menyangkut keberlangsungan bisnis suatu perusahaan, sehingga diperlukan bantuan sistem yang dapat memberikan informasi mengenai statistik jaringannya. Disamping itu, pada kondisi dimana terjadinya masalah pada jaringan, hal ini juga menjadi tanggung jawab administrator. Informasi mengenai permasahan jaringan secara

langsung memiliki dampak pada proses pemulihan jaringan itu sendiri. Semakin cepat informasi tersebut diperoleh, maka proses pemulihan akan semakin cepat pula ditanggulangi.

Nagios sebagai salah satu *network monitroring system* (NMS) memiliki kemampuan untuk mengumpulkan statistik jaringan secara akurat dalam suatu waktu. Nagios juga memiliki kemampuan untuk memberikan pesan peringatan disaat kondisi jaringan sedang bermasalah. Pesan peringatan tersebut dikirimkan dalam bentuk surat elektronik (*electronic mail*). *Email* yang dikirimkan kepada administrator berisi pesan peringatan akan adanya kesalahan pada jaringan, yang dalam hal ini terjadi apabila kondisi statistik jaringan pada suatu waktu melebihi nilai ambang batas yang telah ditetapkan.

Memang notifikasi dengan *email* cukup membantu, terutama untuk pelaporan kepada pihak manajemen yang lebih tinggi. Fasilitas ini akan memberikan informasi kepada administrator yang juga sedang terhubung dengan jaringan tersebut, ataupun minimal memiliki akses dengan media internet untuk dapat menerima pesan masalah jaringan melalui *email*. Namun dibalik itu, masih terdapat kekurangan pada cara pengiriman pesan dengan menggunakan *email*, yaitu:

1. Penerima pesan harus terhubung dengan internet
2. Seorang *network administrator* harus mengetahui pesan peringatan dalam waktu yang seminimal mungkin mengingat jika ada permasalahan jaringan yang berat, informasi mengenai hal itu harus diketahui secara cepat dan akurat untuk mengembalikan kondisi jaringan pada keadaan semula, dengan demikian *email* tidak cukup cepat untuk menangani hal tersebut
3. Fasilitas *email* hanya didukung oleh beberapa jenis telepon seluler
4. Jika fasilitas *email* diakses menggunakan media telepon seluler (untuk selanjutnya disebut HP/ *handphone*), biayanya relatif mahal dan waktu akses yang semakin lama.

Perbandingannya bila Nagios menggunakan media SMS sebagai penyampai pesan notifikasi masalah jaringan adalah:

1. Fitur SMS sudah sangat *familiar*, penggunaannya pun mudah.
2. Reliabilitas dengan semua jenis HP tanpa terkecuali dan penggunaan pulsanya jauh lebih murah dibandingkan fitur lainnya seperti MMS (*Multimedia Message Service*).
3. Pengoperasian SMS relatif lebih mudah dan lebih cepat dibandingkan *email*

Oleh sebab itu, pertimbangan akan dibangunnya fitur baru berbasis SMS yang melengkapi fitur Nagios diharapkan dapat menjadi suatu solusi bagi permasalahan yang ada.

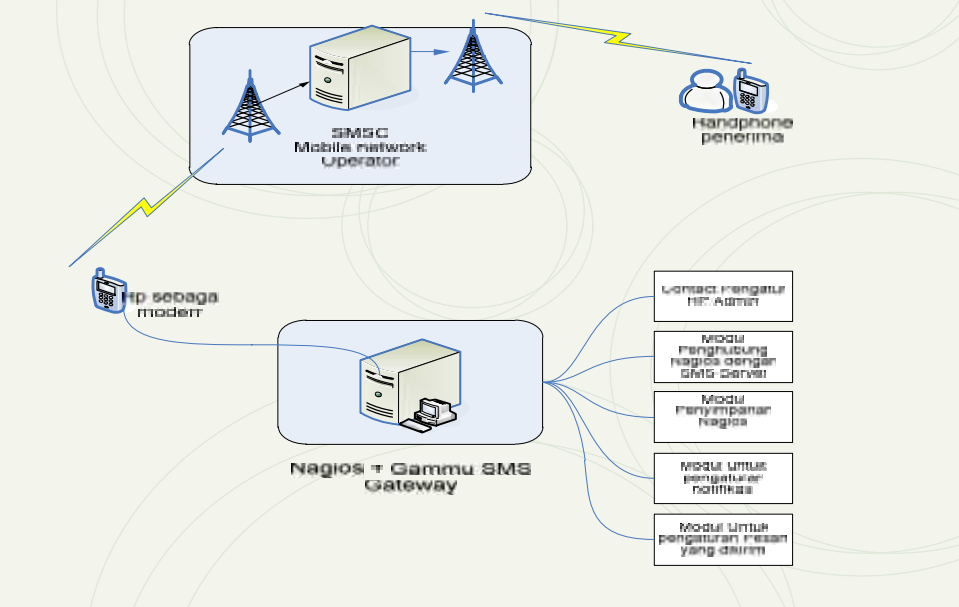
4.2.2. Deskripsi Umum Sistem Yang Akan Dibangun

Sistem yang akan dibangun merupakan rancang bangun pengembangan fitur dari sistem yang sudah ada. Pengembangan dan penambahan fitur yang dimaksud adalah fitur SMS pada Nagios.

Rincian penjelasan rancang bangun fitur tersebut dapat dilihat pada penjelasan dibawah ini:

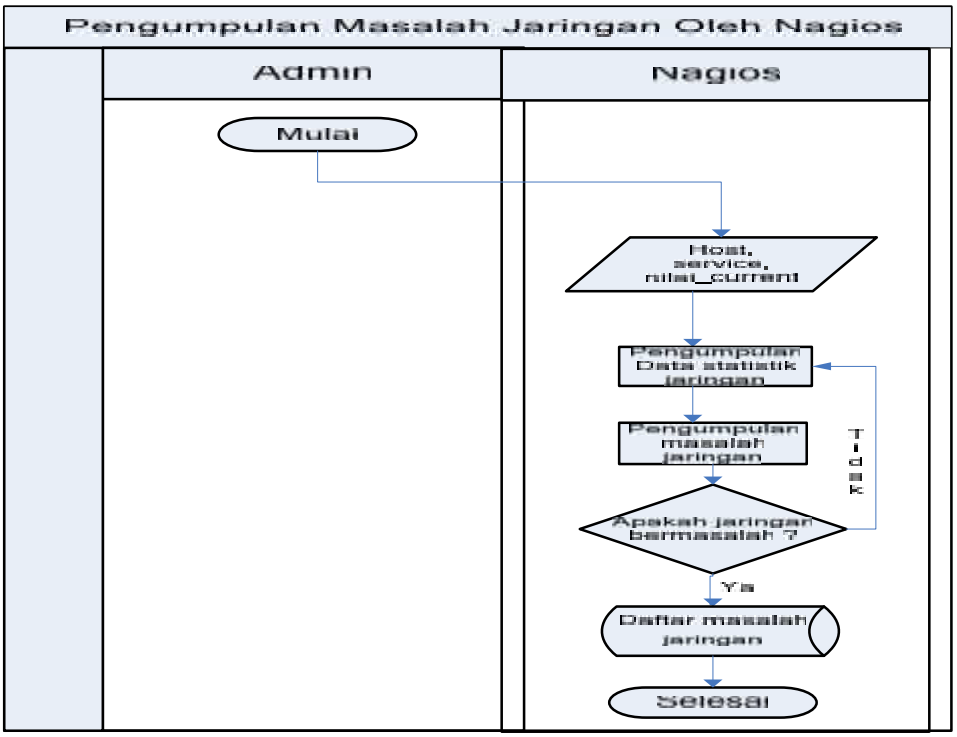
1. Nagios merupakan sistem pemantauan jaringan yang bertugas memantau aktifitas jaringan kemudian Nagios yang akan memberikan peringatan bila tejadi masalah pada jaringan
2. Masalah pada jaringan yang dikumpulkan dan disimpan dalam basis data Nagios selanjutnya akan diproses oleh modul SMS tambahan yang kemudian akan dikirimkan menggunakan HP yang berfungsi sebagai modem.
3. SMS peringatan akan diterima dengan segera oleh HP *network administrator* yang tidak sedang berada di tempat.

Rancang bangun Nagios secara umum dapat dilihat pada gambar 4.2.



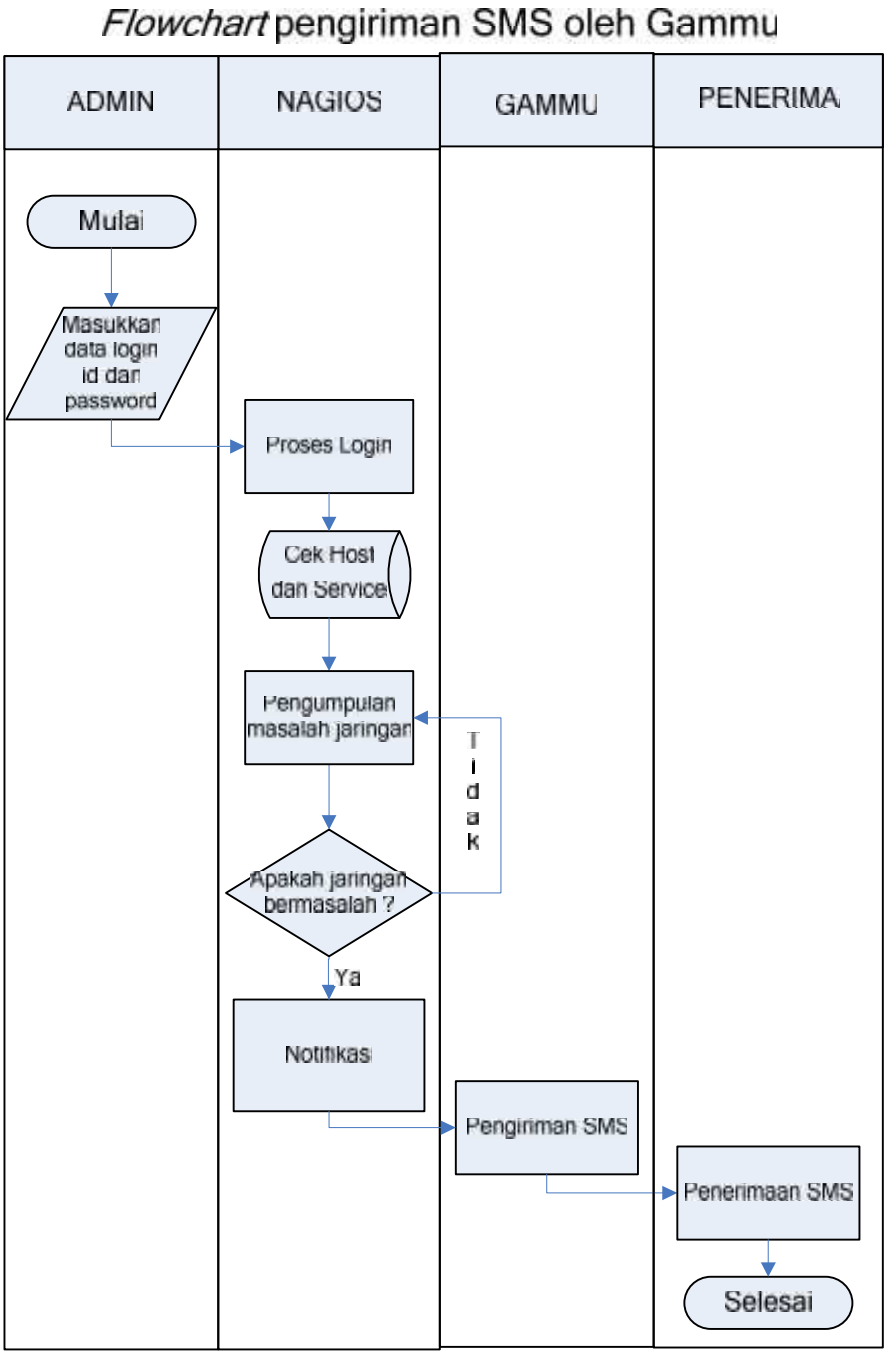
Gambar 4.2 Deskripsi Umum Perangkat Lunak yang akan dibangun

Flowchart berikut menggambarkan proses pengumpulan data masalah jaringan yang dilakukan oleh Nagios.



Gambar 4.3 Proses Pengumpulan Data Masalah Jaringan oleh Nagios

Flowchart berikut merupakan proses pengiriman SMS yang dilakukan oleh perangkat lunak yang akan dibangun



Gambar 4.4 Proses Pengiriman SMS pada Nagios

4.2.3. Analisa Perangkat Lunak

Analisa Perangkat Lunak bermanfaat sebagai penentu proses pengerjaan agar ditemukannya suatu pemecahan masalah. Terciptanya kerunutan suatu analisa perangkat lunak pada jalur yang benar merupakan dasar dilakukannya tahap ini.

Analisa perangkat lunak terdiri atas analisa kebutuhan data, analisa kebutuhan perangkat, analisa kebutuhan fungsi, dan analisa fungsional

4.2.3.1. Analisa Kebutuhan Data

Sistem yang akan dibangun memiliki kriteria kebutuhan data sebagai berikut:

a. Data *host*, berupa IP *host* dan status *host*, ini digunakan untuk menginisialisasi status *host* sekaligus untuk membedakan *host* satu dengan lainnya

b. Data *service* dan status *service*, ini digunakan untuk menginisialisasi status *service* sekaligus untuk membedakan *service* satu dengan lainnya

c. Nilai ambang batas yang telah ditetapkan pada *script*, nilai data ini bersifat fleksibel karena ditetapkan sendiri ambang batasnya oleh *network administrator* dan akan digunakan sebagai pembanding bagi nilai yng sedang diukur pada suatu waktu

Data-data tersebut telah ada dalam basis data Nagios, sehingga yang akan dibuat adalah membangun *script* yang dapat memanggil data-data tersebut untuk dapat dibandingkan dengan nilai-nilai yang ada. Apabila terjadi suatu masalah dijaringan yang dalam hal ini ditandai dengan dilanggarnya nilai ambang batas, maka data akan masuk ke *database* dan kemudian dibandingkan untuk kemudian akan dikeluarkan notifikasi sebagai informasi *alert* berupa SMS.

4.2.3.2.Analisa Fungsi yang akan dibangun pada Perangkat Lunak

Perangkat lunak ini memiliki beberapa fungsi yang bisa dimanfaatkan oleh *user* agar dapat menghasilkan *output* yang maksimal dan berjalan sebagaimana mestinya. Berikut rincian fungsi yang ada didalam fitur yang akan dibangun pada perangkat lunak adalah:

1. *Input host*
2. Pembandingan nilai basis data Nagios pada suatu waktu dengan nilai *Input* pada *host*
3. Pengiriman hasil pembandingan pada nomor 2 dalam bentuk SMS
4. Penyimpanan *log* pesan telah terkirim

BAB V

IMPLEMENTASI DAN PENGUJIAN

5.1. Implementasi Sistem

Implementasi merupakan tahap lanjutan setelah analisa dan perancangan dilakukan. Pada tahapan ini, sistem yang telah selesai, siap untuk dioperasikan dan dilakukan pengujian untuk melihat sejauh mana sistem yang dibuat dapat mencapai tujuan.

Implementasi merupakan kelanjutan dari tahap perancangan sistem yang telah didesain. Pada tahap ini difokuskan kepada penerapan sistem yang didesain pada bahasa pemrograman yang sesuai, sehingga akan diperoleh hasil yang akan diinginkan.

Tujuan implementasi yaitu:

1. Menyelesaikan desain sistem yang ada dalam dokumen perancangan
2. Menguji program-program atau prosedur-prosedur dari dokumen perancangan sistem
3. Mempertimbangkan bahwa sistem sesuai dengan harapan yakni dengan menguji secara keseluruhan.

Langkah-langkah yang dibutuhkan dalam implementasi sistem adalah sebagai berikut:

1. Menyelesaikan desain sistem
2. Memilih Perangkat Lunak
3. Menyiapkan Lingkungan Implementasi
4. Melakukan konversi dari apa yang telah dirancang kedalam bahasa pemrograman (*coding*)
5. Menguji sistem

5.1.1. Alasan Pemilihan Perangkat Lunak

Fitur yang akan dibangun akan menggunakan perangkat lunak berbasis web, yakni PHP. Beberapa pertimbangan digunakannya perangkat lunak tersebut adalah:

1. PHP merupakan *software* yang *open source* (gratis) dan mampu lintas *platform*, yaitu dapat digunakan dengan sistem operasi dan *web server* apapun (windows dan beberapa versi linux)
2. PHP juga merupakan *software* yang dinamis
3. PHP menawarkan konektifitas yang baik dengan berbagai macam basis data
4. PHP memiliki kecepatan dalam eksekusi perintah dan mampu menangani jutaan *request* dalam waktu yang bersamaan

5.1.2. Lingkungan Implementasi

Lingkungan implementasi sistem ada dua yaitu lingkungan perangkat keras (*hardware*) dan lingkungan perangkat lunak (*software*).

5.1.2.1. Perangkat keras (*hardware*)

Perangkat keras yang akan digunakan sebagai tempat untuk instalasi adalah komputer yang memiliki spesifikasi sebagai berikut:

a. *Processor* Pentium 4 (2,0 GHz)

b. *Memory* 512 *MB*

c. *Hardisk* berkapasitas 40 *GB.*

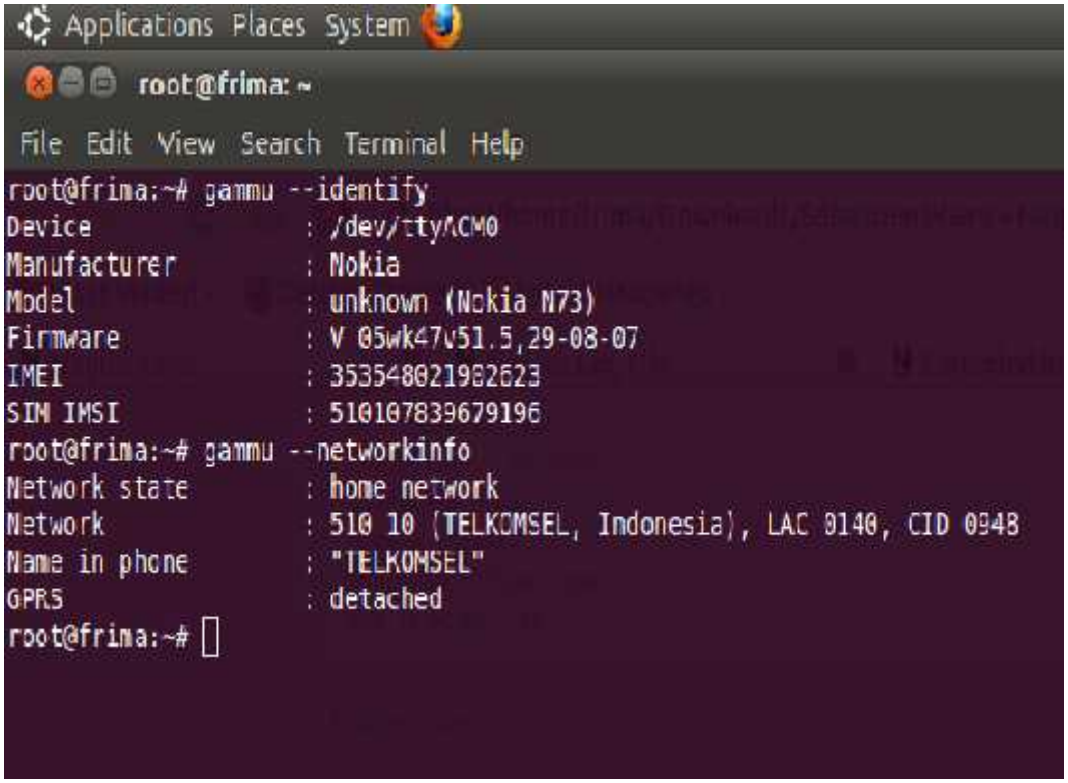
d. *Network Interface Card* (NIC)/Kartu jaringan

Perangkat keras pendukung yakni modem yang berupa *mobile phone* memiliki spesifikasi yakni HP Nokia N73 dengan kabel penghubungnya. Proses konfigurasi menggunakan Gammu SMS *gateway*. Caranya adalah hubungkan *port*

serial USB pada HP nokia kemudian Gammu SMS *gateway* untuk melakukan konfigurasi. Edit *script* Gammurc untuk mengenalkan *port serial* HP pada Gammu SMS *gateway*. Pada *script* Gammurc, pilih *port = /dev/ttyS0 connection* = at115200.

Gambar 5.1 Proses *Edit script* Gammurc

Kemudian lakukan test modem dengan # gammu --*identify*



Gambar 5.2 Proses Test Modem Pada Gammu

5.2.1.2 Perangkat lunak (*software*)

Lingkungan perangkat lunak untuk implementasi aplikasi ini adalah Sistem:

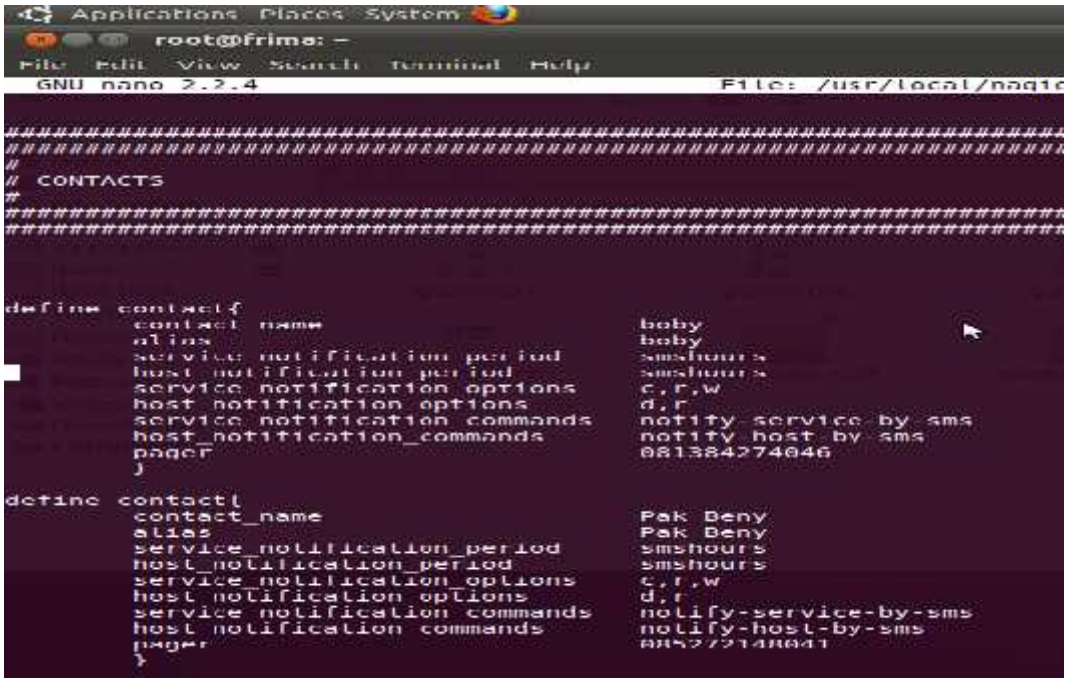
1. Sistem Operasi : *Linux Ubuntu 10.10*
2. *Web Server* : *Apache* 2.2
3. *Web browser* : *Firefox Mozilla* 4.0.

5.1.3. Hasil Implementasi

Implementasi dari Nagios dan Gammu menghasilkan beberapa tambahan script baru untuk mengatur proses pengiriman SMS berisi notifikasi.

5.1.3.1.Script Contact

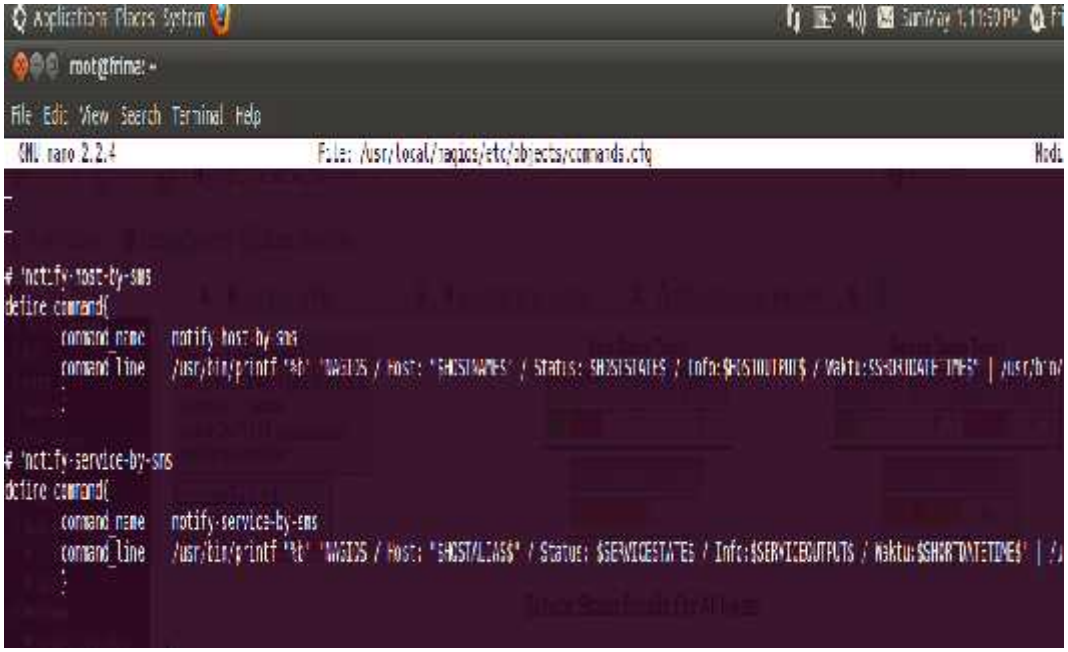
Script contact merupakan *script* yang berfungsi untuk mengatur nama *contact* dan nomor HP yang akan menerima pesan. Sehingga kita dapat menentukan siapa saja yang di tetapkan menerima notifikasi dari Nagios nantinya.



Gambar 5.3 *Contain Script Contact*

5.1.3.2.Script Command

Script yang mengatur *command line* dan isi pesan yang akan dikirimkan oleh Nagios nantinya. Sehingga isi pesan dapat kita buat sesuai dengan yang kita inginkan.



Gambar 5.4 *Contain Script Command*

5.2. Pengujian Sistem

Tahap ini merupakan tahap terakhir dari proses pembuatan aplikasi ini. Setelah tahap implementasi dilakukan maka aplikasi harus diuji agar dapat diketahui hasil dari program implementasi sistem. Pengujian dilakukan dengan tujuan untuk menjamin bahwa sistem yang dibuat sesuai dengan hasil analisa, perancangan dan tujuan sebelumnya, juga untuk menemukan kesalahan yang mungkin terjadi sehingga menghasilkan suatu kesimpulan.

Pengujian sistem dilakukan dengan *black box* berfokus dan pengujian akurasi data. Pengujian *black box* berfokus adalah istilah yang digunakan untuk menjelaskan berlangsungnya pengujian saat merancang uji kasus sehingga pengujian tersebut dapat diterima. Pengujian akurasi data adalah pengujian yang dilakukan untuk membuktikan apakah diperoleh kesesuaian data yang dicari

secara manual dengan data yang diproses dari sistem. Bila sesuai maka pengujian tersebut telah berhasil.

Keterangan format yang digunakan pada tabel-tabel yang berisikan butir-butir pengujian yaitu:

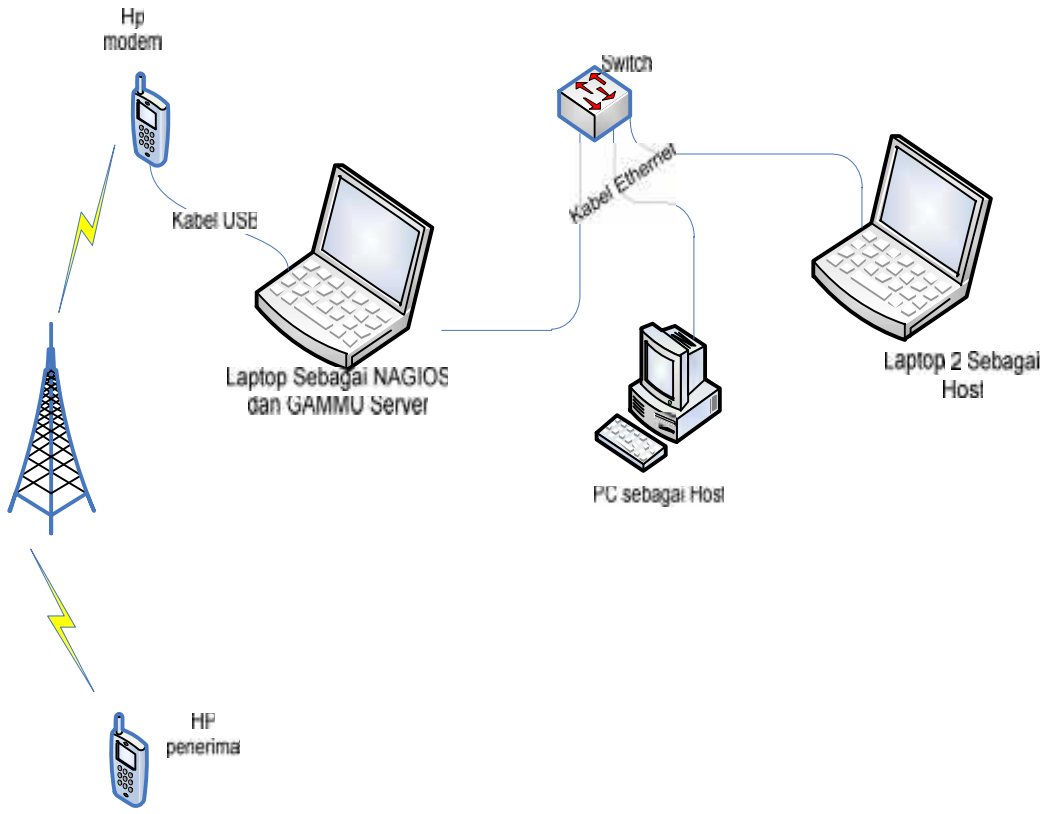
1. Deskripsi, menerangkan pengujian pada apa yang akan dilakukan
2. Prosedur pengujian, merupakan langkah-langkah untuk melakukan pengujian yang dideskripsikan
3. Masukan, yaitu *input* data yang akan diolah
4. Keluaran yang diharapkan, yaitu kesuksesan dari data yang diolah
5. Kriteria evaluasi hasil, yaitu apakah hasil yang diinginkan sesuai dengan keinginan pengguna
6. Hasil yang didapat, menerangkan apakah sistem tersebut berhasil atau tidak
7. Kesimpulan, yaitu diterima atau tidaknya hasil dari pengujian

5.2.1. Kebutuhan Perangkat Uji Coba dan Skenario Pengujian

Pada pembahasan analisa, telah disinggung mengenai kriteria prioritas suatu masalah yang terjadi di jaringan. Dalam bahasan ini, akan dipaparkan mengenai kebutuhan perangkat uji coba yang akan digunakan.

Perangkat yang akan digunakan sebagai *Host* untuk uji coba merupakan perangkat keras yang dapat mewakili masing-masing kriteria dari masalah jaringan. Sebagai contoh, sebuah Laptop yang dapat dianalogikan sebagai *server* dengan fitur-fitur yang diperlukan telah di-*install* didalamnya seperti Nagios, Gammu dan PHP. Beberapa perangkat lain yang dipakai adalah sebuah laptop dan sebuah *Switch* 8 *port*. perangkat tersebut cukup untuk mewakili masing-masing kriteria resiko permasalahan jaringan.

Berdasarkan penjelasan diatas maka dapat disusun suatu skenario pengujian utnuk membuktikan apakah sistem yang dibangun memiliki hasil sesuai dengan yang diharapkan atau tidak. Skenario pengujian tertuang pada gambar 5.5.



Gambar 5.5 Skenario Pengujian

Dari gambar dapat dijelaskan bahwa sebuah laptop akan difungsikan sebagai Nagios sekaligus Gammu *server*. Sebuah laptop dan PC dipakai sebagai *Host* yang akan diuji. Sebuah HP yang dipasang menggunakan kabel USB pada laptop berfungsi sebagai modem. Penghubung antara Laptop Nagios dan switch adalah kabel *ethernet*.

Pengujian tersebut telah dilaksanakan pada tanggal 25 April 2011 di salah satu warung internet di Pekanbaru.

5.2.2. Pengujian Proses *Login* Pada Nagios

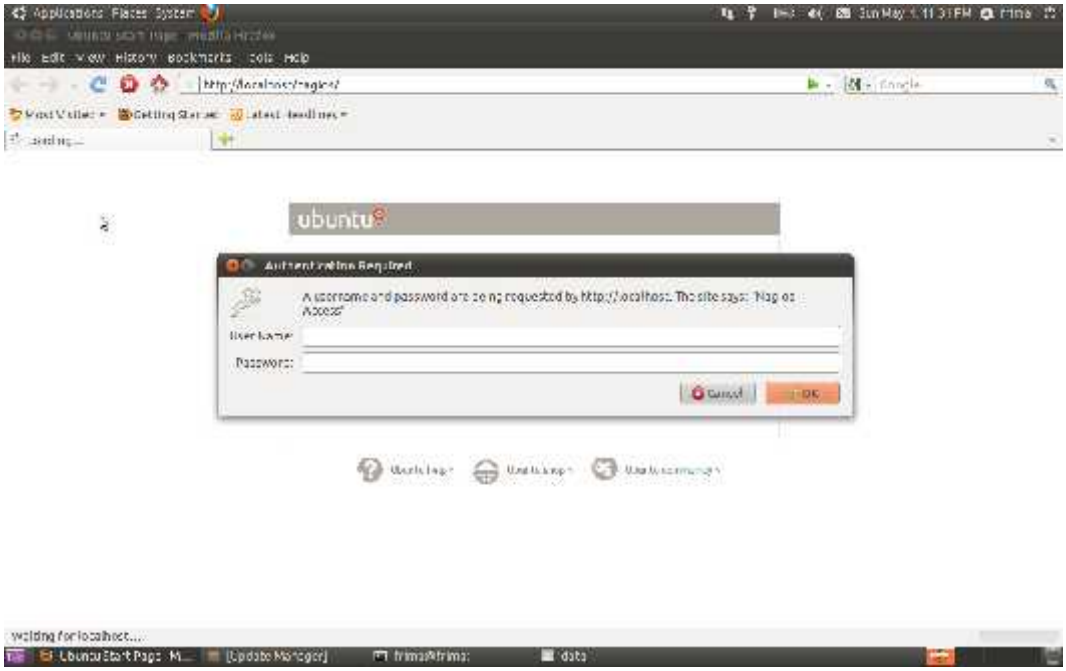
Jika *Login* pada Nagios berhasil maka administrator memiliki hak akses penuh terhadap Nagios.

Prekondisi: Halaman Awal *Login*

Tabel 5.1 Butir Pengujian *Login* Nagios

Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang Diharapkan	Kriteria Evaluasi Hasil	Hasil yang Didapat	Kesimpulan
Pengujian *Login*	1. pada *web browser,* ketikkan http//: *localhost*/ nagios lalu *enter* *2.* ketikkan *username* dan *password*	*Username* dan *password*	*Login* Nagios berhasil.	*Login* Nagios berhasil.	*Login* Nagios berhasil. Dapat mengakses Nagios.	Diterima

Hasil dari pengujian dapat dilihat pada gambar 5.6 dan gambar 5.7.



Gambar 5.6 Proses *Login* Nagios

Tampilan setelah *username* dan *password* dimasukkan adalah menu utama Nagios.

Gambar 5.7 Tampilan Menu Utama Nagios

5.2.3. Pengujian Pendaftaran *Host* pada Nagios

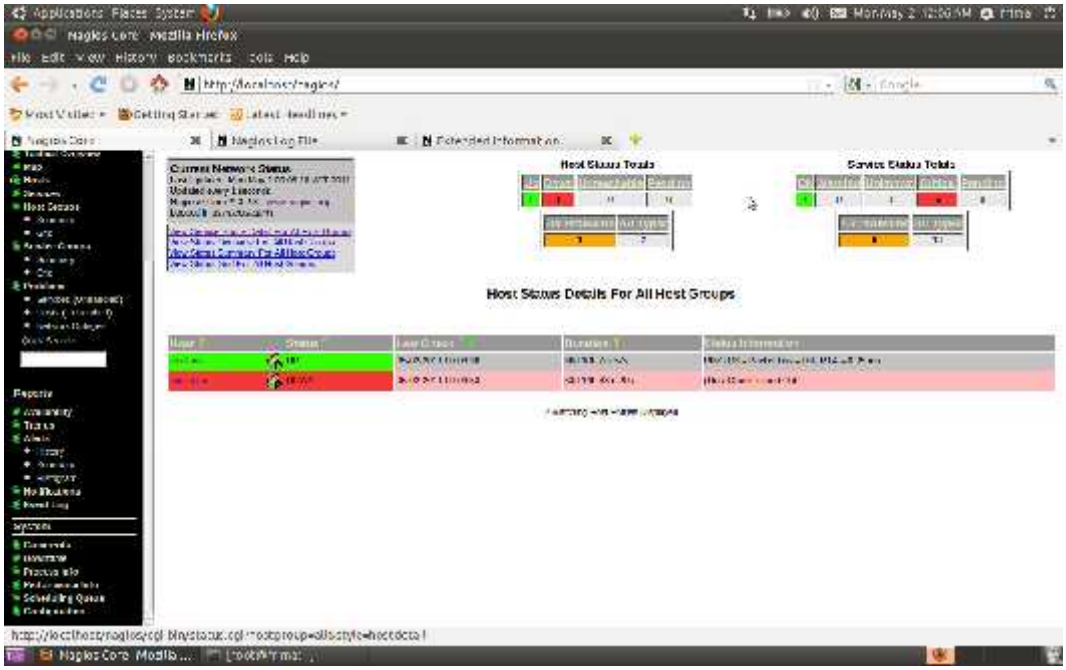
Agar masalah suatu *Host* dapat didefenisikan dan dikirimkan, hal paling utama dilakukan adalah mendaftarkan *Host* tersebut dalam Nagios. Proses pendaftaran ini nantinya akan memberikan inputan bagi Nagios. Sebagai contoh, mendaftarkan *localhost*.

Prekondisi: Berada Terminal Linux, *script* Nagios dan *script Host*.

Tabel 5.2 Butir Pengujian Pendaftaran *Host*

Deskripsi	**Prosedur Pengujian**	**Masukan**	**Keluaran yang Diharapkan**	**Kriteria Evaluasi Hasil**	**Hasil yang Didapat**	**Kesimpulan**
Pengujian Pendaftaran *Host*	1. Buka terminal pada linux. 2. *Login* ke *super user*. 3. Tambahkan script *Host* yang kita inginkan pada *folder* 4. *Edit script* nagios dan tambahkan nama *script* yang sebelumnya kita tambahkan pada *folder* pada *script* nagios. 5. *Restart* nagios.	1. *Edit script Host* sesuaikan *ip addres* dan lainnya. 2. *Edit script* Nagios dan tambahkan nama *script Host* tersebut didalamnya.	*Host* berhasil didaftarkan pada Nagios	*Host* berhasil didaftarkan pada Nagios	*Host* Terdaftar	Diterima

Untuk hasilnya dapat dilihat pada gambar 5.8



Gambar 5.8 Contoh Hasil Pengujian Pendaftaran *Host*

5.2.4. Pengujian Pendaftaran *Contacts*

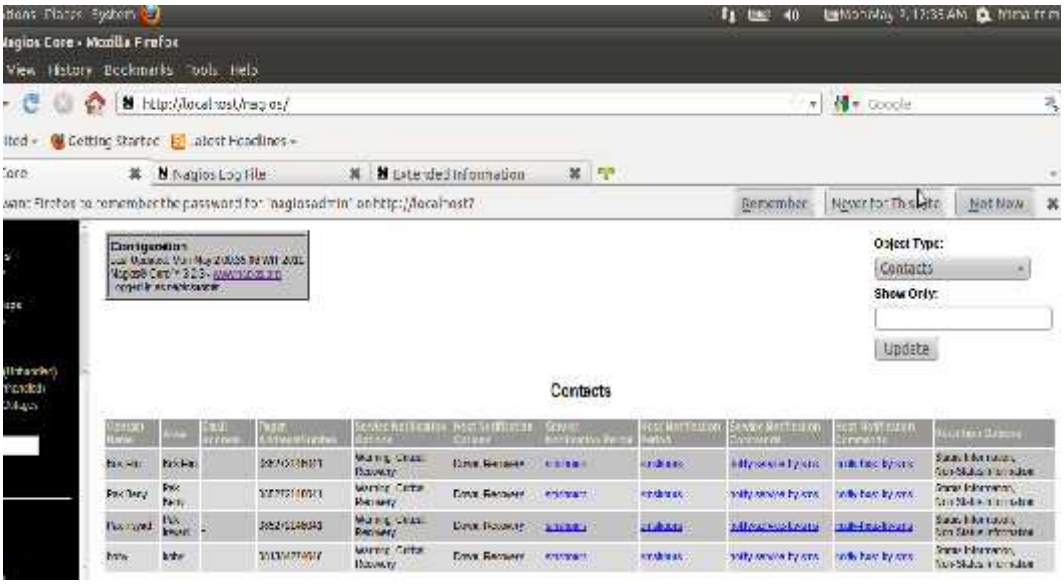
Setelah *Host* didaftarkan, saatnya *contact* juga didaftarkan.

Prekondisi: berada pada terminal linux, *script contact*

Tabel 5.3 Butir Pengujian Pendaftaran *Contact*

Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang Diharapkan	Kriteria Evaluasi Hasil	Hasil yang Didapat	Kesimpulan
Pengujian Pendaftaran *Contact*	1. Buka terminal pada linux. 2. *Login* ke *super user*. 3. *Edit script contact*, masuk kan daftar *contact* yang kita inginkan. 4. *Restart* nagios.	Nama *contact,* nomor HP	*Contact* berhasil ditambah	*Contact* berhasil ditambah	*Contact* bertambah	Diterima

Untuk mengetahui hasil uji coba penambahan *contact*, dapat dilihat pada gambar 5.9



Gambar 5.9 Hasil Uji Coba Penambahan *Contact*

5.2.5. Pengujian Gammu SMS *gateway*

Pengujian pada fitur Gammu SMS *gateway* melalui dua tahapan pengujian yakni pengujian identifikasi modem atau HP, dan pengiriman isi pesan melalui SMS.

5.2.5.1. Pengujian identifikasi modem/HP

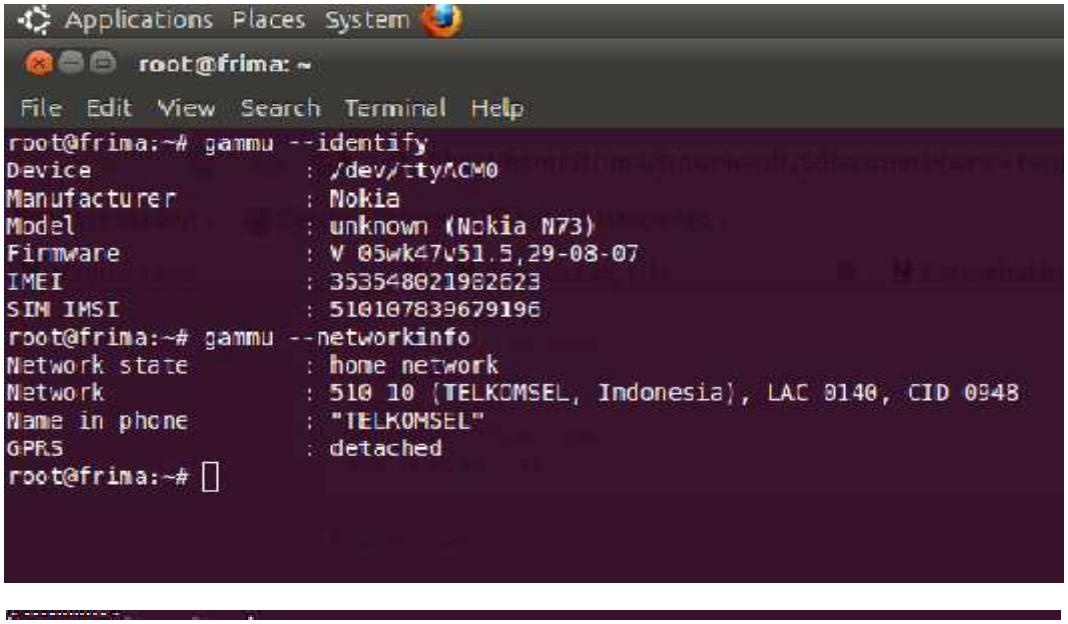
Prekondisi: berada pada terminal linux, *folder* gammu

Tabel 5.4 Butir Pengujian identifikasi modem/HP

Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang Diharapkan	Kriteria Evaluasi Hasil	Hasil yang Didapat	Kesimpulan
Pengujian identifikasi modem/HP	1. Buka terminal pada linux, *Login* sebagai *super user*. 2. Masuk ke *folder* Gammu	Data modem/HP yang *support* pada Gammu.	1. Nama modem/HP serta keterangan *support* pada Gammu.	1. Nama modem/HP serta keterangan *support* pada Gammu.	Info atau keterangan modem/HP yang telah berhasil di identifikasi	Diterima

	3. Kemudian identifikasi modem dengan mneggunakan *syntax identify* pada Gammu.		2. Keterangan bahwa modem/HP telah berhasil di identifikasi oleh Gammu	2. Keterangan bahwa modem/HP telah berhasil di identifikasi oleh Gammu	Gammu	

Hasilnya sama seperti yang terdapat pada gambar 5.10



Gambar 5.10 Pengujian identifikasi modem/ HP

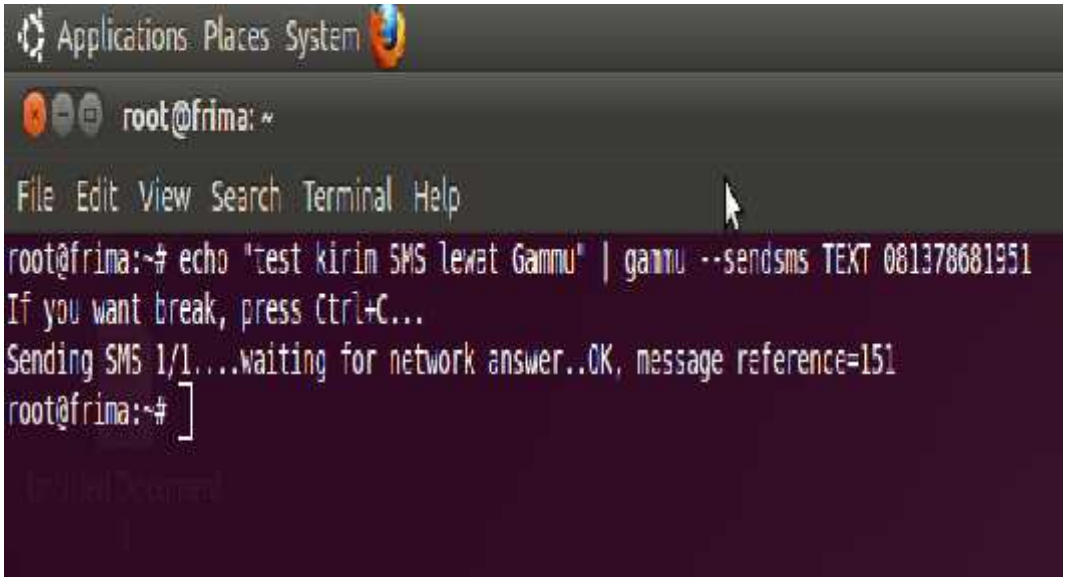
5.2.5.2.Pengujian pengiriman isi SMS pada Gammu

Prekondisi: berada pada terminal linux, *folder* gammu

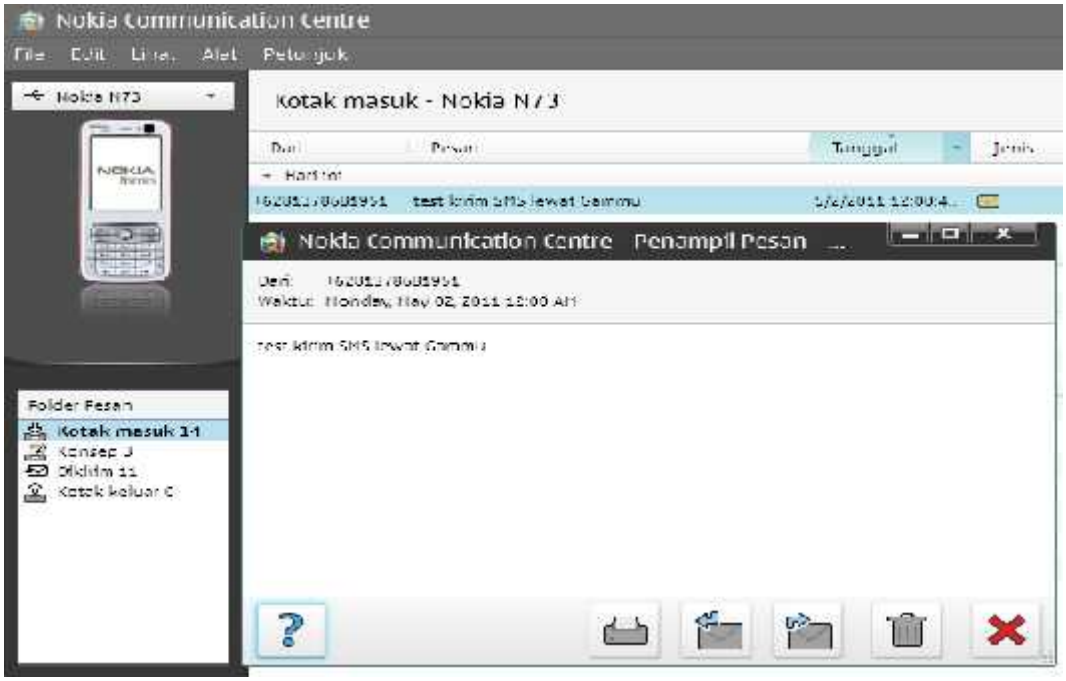
Tabel 5.5 Butir Pengujian Manajemen HP

Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang Diharapkan	Kriteria Evaluasi Hasil	Hasil yang Didapat	Kesimpulan
Pengujian pengiriman isi SMS pada Gammu	1. Buka terminal pada linux, *Login* sebagai *super user*. 2. Masuk ke *folder* Gammu 3. Kemudian kirim pesan dengan *syntax echo* pada gammu.	Isi pesan SMS yang ingin dikirim dan nomor HP.	Isi pesan SMS terkirim ke nomor yang telah ditetapkan.	Isi pesan SMS terkirim ke nomor yang telah ditetapkan.	Isi pesan SMS terkirim ke nomor yang telah ditetapkan.	Diterima

Hasilnya sama seperti yang terdapat pada gambar 5.11



Gambar 5.11 Pengujian isi SMS pada Gammu di Linux



Gambar 5.12 Pengujian isi SMS pada Gammu di Layar HP

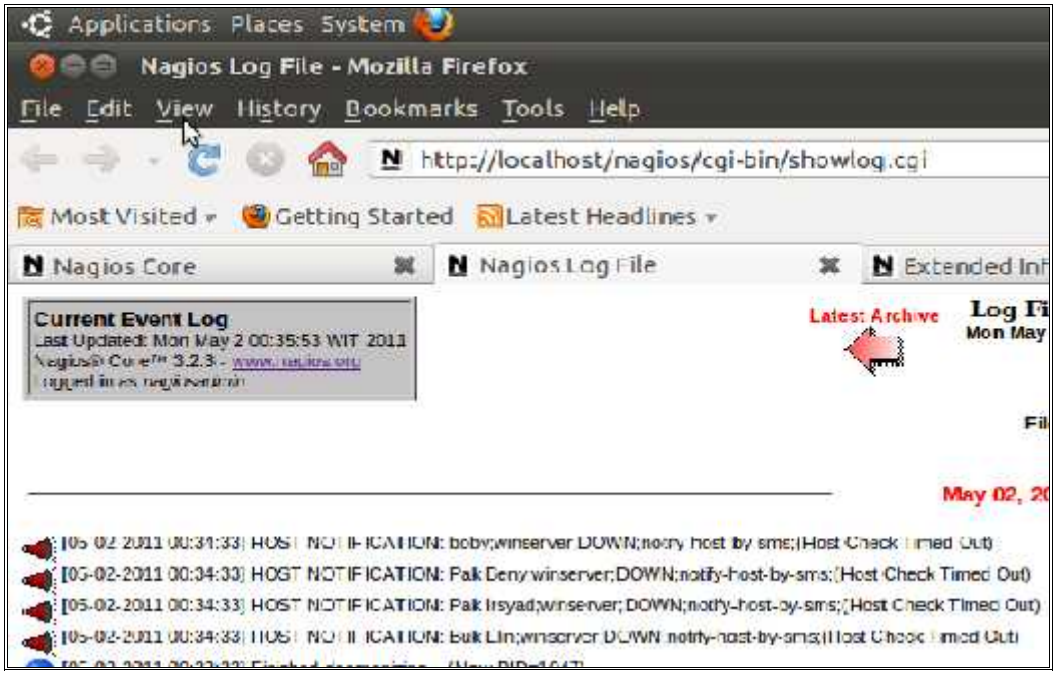
5.2.6. Pengujian Pengiriman SMS Pada Fitur Nagios

Prekondisi: notifikasi *Host* dan *service* pada Nagios

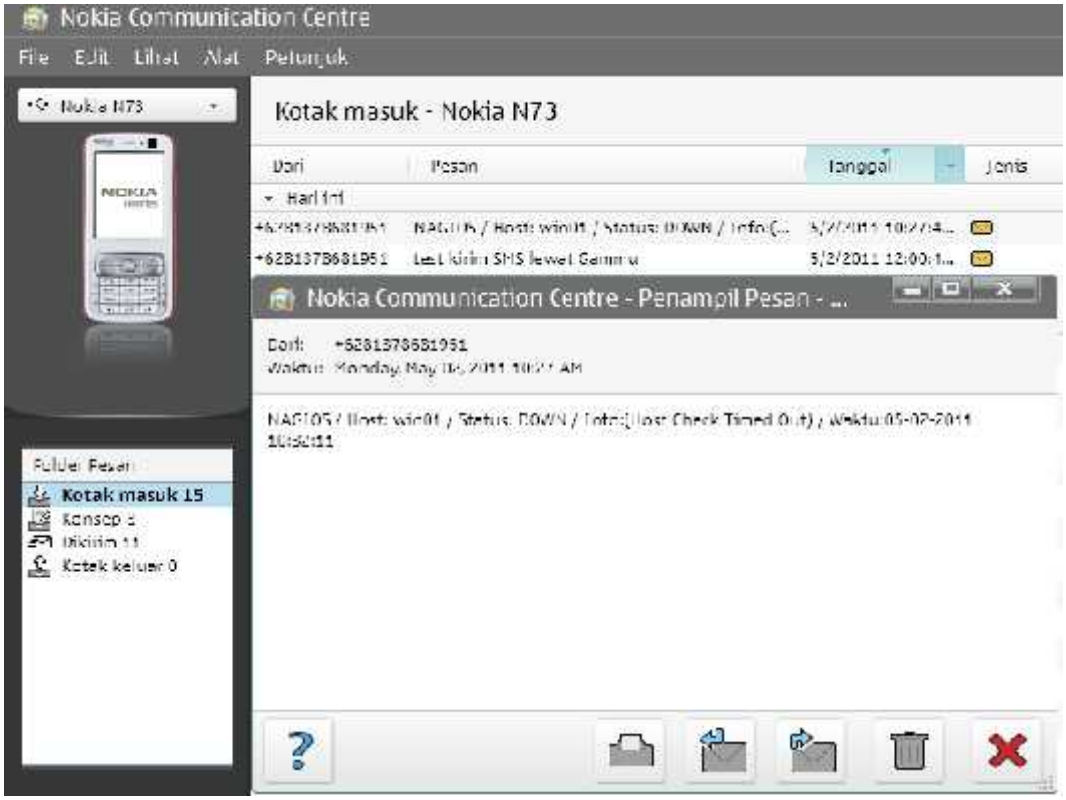
Tabel 5.6 Butir Pengujian Pengiriman SMS Pada Fitur Nagios

Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang Diharapkan	Kriteria Evaluasi Hasil	Hasil yang Didapat	Kesimpulan
Pengujian Pengiriman SMS	1. Jalankan Nagios 2. Pasang semua *hardware* pendukung. Sesuai dengan *Host* dan *service* yang telah terdaftar di Nagios 3. Lakukan notifikasi.	1. Pilihan *Host* 2. Pilihan *service*	Notifikasi *Host* dan *service* yang dan tampil dilayar SMS pada HP	Notifikasi *Host* dan *service* yang dan tampil dilayar SMS pada HP	Notifikasi *Host* dan *service* yang dan tampil dilayar SMS pada HP	Diterima

Untuk melihat hasil pengujian, dapat dilihat pada gambar 5.13 dan 5.14



Gambar 5.13 Hasil Pengujian Pengiriman SMS Pada Fitur Nagios Info *Log*



Gambar 5.14 Hasil Pengujian Pengiriman SMS Pada Fitur Nagios Layar HP

5.2.7. Pengujian *Poller* Nagios

Prekondisi: *Poller* Nagios

Tabel 5.6 Butir Pengujian *Poller* Nagios

Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang Diharapkan	Kriteria Evaluasi Hasil	Hasil yang Didapat	Kesimpulan
Pengujian Waktu *Poller* Nagios	1. Jalankan Nagios 2. Menghitung Waktu yang diperlukan dalam proses pengiriman *alert* lewat SMS diterima oleh *Contacts*	1. Pilihan *Host* dan *service*	Waktu Notifikasi *Host* dan *service* lewat SMS diterima oleh HP *Contacts* maksimal 1 menit	Waktu Notifikasi *Host* dan *service* lewat SMS diterima oleh HP *Contacts* maksimal 1 menit	Waktu Notifikasi *Host* dan *service* lewat SMS diterima oleh HP *Contacts* kurang dari 1 menit	Diterima

5.2.8. Kesimpulan Hasil Pengujian

Setelah dilakukan tahap pengujian terhadap sistem, maka dapat ditarik kesimpulan berkenaan dengan hasil pengujian, yaitu:

1. *Network* Administrator memiliki hak akses terhadap Nagios.
2. *Script* pengisian data yang terdapat pada *script Command, Contact*, Nagios, Gammurc dan *Host* yang ada dapat berjalan sesuai dengan yang diharapkan
3. Sms peringatan berhasil dikirimkan, dan pesan hanya dapat terkirim bila nomor HP penerima didaftarkan pada *script Contact*.
4. SMS terkirim secara otomatis tanpa harus meng-klik tombol *refresh* setiap saat.
5. Pesan yang terkirim hanyalah pesan dengan notifikasi *alert* pada *Host* dan *Service.*
6. *Poller* implementasi Nagios maksimal selama 1 menit, mendekati *realtime*.

BAB VI

PENUTUP

Bab ini merupakan bab penutup yang berisi kesimpulan dari tujuan awal dan saran yang diperlukan untuk mengembangan Fitur Nagios berbasis SMS selanjutnya.

6.1 Kesimpulan

Berdasarkan pada pembahasan pada Bab-Bab sebelumnya, dapat diambil suatu kesimpulan yakni:

1. Notifikasi berbasis SMS merupakan fitur baru yang diintegrasikan dari Nagios dan Gammu untuk mengatasi kekurangan Nagios sebelumnya, fitur ini dapat menyampaikan *alert* dalam bentuk SMS
2. *Poller* implementasi Nagios maksimal selama 1 menit, mendekati *realtime.*
3. *Script* pengisian data yang terdapat pada *script Command, script Contact*, *script* Nagios, *script* Gammurc dan *script Host* yang ada dapat di integrasikan sesuai tujuan.

6.2 Saran

Aplikasi yang dibangun ini masih memiliki kekurangan, untuk itu pada bagian ini akan dikemukakan beberapa kekurangan untuk kemudian diperbaiki dan dikembangkan, seperti:

1. Nagios hanya mampu untuk mengirimkan SMS searah, fasilitas untuk *trouble shooting* jaringan secara *remote* belum tersedia sehingga perlu dikembangkan sistem yang mampu mengatasi kekurangan tersebut
2. Nagios berbasis SMS (Gammu) memiliki keterbatasan dalam hal mengetahui serangan penyusup di jaringan, oleh sebab itu perlu diintegrasikan dengan sistem IDS (*Intrusion Detection System*) agar lebih *powerfull* dan dapat menangkap seluruh penyebab terjadinya masalah di jaringan

3. Sistem belum mampu untuk mengetahui kondisi sinyal operator dan jumlah pulsa modem secara otomatis

DAFTAR PUSTAKA

Arisanti, Eky Rahayu. Tugas Akhir *Rancang Bangun Fitur Cacti Berbasis Short Messages Service Untuk Penyampaian Notifikasi Masalah Jaringan Dengan Priority Service*, Pekanbaru 2010

Ekklesya. *Network Trouble Shooting* [*Online*] *Available* http://ekklesya. blogspot.com/2009/12/network-trouble-shooting.html, diakses 21 Januari 2011

Hendradhy, Oke. *Strategi Pengembangan Manajemen Resiko dalam Pengembangan Sistem Informasi.* [*Online*] *Available* http://mugi.or.id, diakses 24 Januari 2011

Jogiyanto. *Analisa dan Desain Sistem Teknologi Informasi.* Andi Yogyakarta. Yogyakarta: 1999.

Levin, Richard I, dkk. *Quantitative Approaches to Management (Seventh Edition)*. McGraw – Hill, Inc. New Jersey: 2002.

Pressman, Roger S. *Software Engineering: A practitioner Approach.* The Mc Graw Hill Companies, Inc. New Jersey: 2005

Sysneta. *Standar Keamanan Jaringan Komputer.* [*Online*] *Available* www.sysneta.com/standar-keamanan-jaringan.html, diakses 4 januari 2011

Team, Cisco. SNMP *Tutorial* [*Online*] *Available* http://www.cisco.com/en/US/docs/internetworking/technology/handbook/ SNMP.pdf, diakses 9 Maret 2011

Terplan, Kornel. *Communication Network Management, 2nd Edition.* Prentice Hall. Englewood Cliffs, New Jersey: 1992.

Yodi, *Dasar untuk Membuat Jaringan Komputer yang Aman.* [*Online*] *Available* http://yodi.web.ugm.ac.id diakses 2 Januari 2011

_____*Download* Nagios *and Plug In* [*Online*] *Available* www.nagios.org Diakses 10 Februari 2011

_____Manajemen Jaringan – Sebuah Tinjauan [*Online*] *Available* http://noegroz.wordpress.com/2007/07/24/manajemen-jaringan-sebuah-tinjauan/, diakses 20 Maret 2011